HUGIN

RUNTIME

USER'S GUIDE

Version 3.1 Revision 2.4

HUGIN Runtime User's Guide

Copyright © 1990–1993 by Hugin Expert A/S.

All Rights Reserved.

This manual was prepared using the LaTeX Document Preparation System by Leslie Lamport running on top of TeX by Donald E. Knuth. Transformed to PostScript using `dvips` by Tomas Rokicki and Donald E. Knuth.

Created by Per Abrahamsen, Lars Peter Fischer, and Lars Bo Nielsen.

Final version prepared by Lars Bo Nielsen.

Version 3.1 Revision 2.4, of January 5, 1994.

Printed in Denmark.

UNIX and OPEN LOOK are registered trademarks of AT&T.

PostScript is a registered trademark of Adobe Systems, Inc.

Sun-3, Sun-4, SPARC, SPARCstation, SPARCprinter, and SunOS are trademarks of Sun Microsystems, Inc.

The X Window System is a trademark of Massachusetts Inititute of Technology.

Hugin Expert, the Hugin Raven Logo, Hugin API, Hugin Regular, and Hugin Runtime are trademarks of Hugin Expert A/S.

Contents

I Tutorial **1**

1 Starting Hugin Runtime **2**

1.1 Before You Start 2

1.2 Getting Started 3

1.3 Choosing a Directory 4

1.4 Choosing a CPN 6

2 The Chest Clinic **8**

3 Creating Monitor Windows **11**

4 Experimentation **14**

4.1 Entering and Propagating Findings 14

4.2 Initializing and New Evidence 16

4.3 Simultaneous Findings 17

4.4 Retracting Findings and Correcting Errors 19

5 A Note on CPN Development **22**

6 The On-line Help System **24**

7 Error Handling **26**

8 Leaving Hugin **27**

II Reference Manual 28

9 The Base Window 29

9.1 The Base Window Pane . . . 30
9.1.1 Selecting . . . 30
9.1.2 Operations . . . 30
9.1.3 Direct Manipulation . . . 30
9.1.4 The Pane's Pop-up Menu . . . 31
9.2 The Mode Setting . . . 31
9.2.1 Click SELECT . . . 32
9.2.2 Dragging . . . 32
9.2.3 Operations . . . 32
9.3 The File Menu . . . 32
9.4 The View Menu . . . 33
9.5 The Edit Menu . . . 34
9.6 The Properties Menu . . . 34

10 The CPN Browser 36

10.1 The Directory Item . . . 36
10.2 The CPN Item . . . 37
10.3 The CPN Buttons . . . 38

11 The Monitor Windows 39

12 The CPN Properties 41

12.1 The Property Items . . . 41
12.1.1 Name . . . 41
12.1.2 Label . . . 42
12.1.3 Width . . . 42
12.1.4 Height . . . 42
12.1.5 Scale . . . 42
12.2 The Apply Button . . . 42
12.3 The Reset Button . . . 42
12.4 The Settings Menu . . . 43

13 The Node Properties 44

13.1 Name and Label . . . 45

13.1.1 Name . . . 45

13.1.2 Label . . . 45

13.2 Parent State Combinations . . . 45

13.3 Adaptation . . . 46

13.4 Weight . . . 46

13.5 The Node States . . . 47

13.5.1 Insert State . . . 47

13.5.2 Delete State . . . 47

13.5.3 State Name . . . 47

13.5.4 State Slider . . . 47

13.5.5 State Probability . . . 47

13.6 Apply, Set Default, and Reset . . . 48

14 The Learning Window 49

14.1 Adaptation . . . 50

14.2 The Batch File . . . 50

14.3 The Learning List . . . 50

14.4 The Generate Item . . . 50

14.5 The Count Item . . . 50

14.6 The Buttons . . . 51

14.7 The Learning View . . . 51

15 The Junction Tree 52

15.1 Layout of the Tree . . . 53

15.2 Selecting Cliques . . . 53

15.3 The Nodes Scrolling List . . . 54

15.4 Node Cliques and Findings . . . 54

15.5 Propagation and Conflicts . . . 56

15.5.1 Definition of Conflict . . . 56

15.5.2 Using Conflicts . . . 56

16 The Case Browser 58

16.1 The Case Item . . . 58

16.2 The Case Buttons . . . 59

17 The Printer Window 60
17.1 Printing on a Printer . . . 60
17.2 Printing to a File . . . 61
17.3 Other Printer Options . . . 62
17.4 Own Size of Output . . . 62
17.5 Previewing . . . 63

18 The Likelihood Window 64
18.1 The Nodes Items . . . 65
18.2 The Weight Item . . . 65
18.3 The Likelihood Values . . . 65
18.4 The Apply Button . . . 65

19 The Tool Properties 66
19.1 The Version Category . . . 67
19.2 The Compilation Category . . . 67
19.2.1 Triangulation . . . 67
19.2.2 Compress . . . 68
19.2.3 Approximate . . . 69
19.3 The Monitor Category . . . 69
19.4 The Pane Category . . . 70
19.4.1 Grid . . . 70
19.4.2 Damping Factor . . . 70
19.4.3 Font . . . 71
19.5 The CPN Browser Category . . . 71
19.6 The Miscellaneous Category . . . 71
19.6.1 CPN . . . 71
19.6.2 Error . . . 72
19.6.3 Learning . . . 72
19.7 The Settings Menu . . . 72

20 Help, Errors, and Other Information 73
20.1 Help . . . 73
20.2 Error Messages . . . 74
20.3 CPN Status . . . 74
20.3.1 Build Mode and Modify Mode . . . 74
20.3.2 Test Mode . . . 75
20.4 Compilation Log . . . 75
20.5 The Demonstration Directory . . . 76

III Appendices 77

A The Built-in Examples 78

A.1 The Chest Clinic Example . . . 78
A.2 The Reproduction Example . . . 79
A.3 The Case of Aunt Emily . . . 81
A.4 The Automobile Example . . . 85

B The CPN Specification Language 87

B.1 The Directory Structure . . . 88
B.2 Specifying Nodes . . . 88
B.3 Specifying Nodes with Parents . . . 89
B.4 Nodes with several Parents . . . 90
B.5 The Complete Specification . . . 91
B.6 Syntax of the CPN Language . . . 93

C Other Fileformats 95

C.1 The Numbering File . . . 95
C.2 The Cases Files . . . 96
C.3 The Adaptation File . . . 96

D OPEN LOOK 98

D.1 The Concepts behind OPEN LOOK . . . 98
D.2 The Mouse . . . 99
D.3 The OPEN LOOK Controls . . . 99
D.3.1 Buttons . . . 100
D.3.2 Menus . . . 101
D.3.3 Exclusive Settings . . . 102
D.3.4 Scrollbars . . . 102
D.3.5 Scrolling Lists . . . 103
D.3.6 Check Boxes . . . 104
D.3.7 Sliders . . . 104
D.3.8 Text Fields . . . 104
D.4 Splitting a Pane . . . 105
D.5 Window Scaling . . . 106
D.6 The OPEN LOOK Window Manager . . . 107
D.6.1 Moving a Window or Icon . . . 107
D.6.2 Resizing a Window . . . 107

D.6.3 Closing a Window and Opening an Icon 108
D.6.4 Dismissing a Pop-up Window 108

E The X Window System and Window Managers 110
E.1 Window Placement 110
E.2 Icons and Closed Windows 111
E.3 Pinned Windows 112

Preface

Hugin Runtime is a development environment for *causal probabilistic networks* (CPNs), also known as *belief networks* or *influence diagrams*. You can create, edit, and test CPNs using Hugin Runtime. CPNs developed with Hugin Runtime can be integrated into your own applications, by using the Hugin API [FA91] (*Hugin Application Programmer Interface*), available as a separate product.

The first part of this manual is a tutorial, explaining at the same time what a CPN is, and how to use Hugin Runtime for testing a CPN. The second part of the manual is a complete reference for Hugin Runtime, covering all the features of the program. The reference part is intended to be short and precise. The final part of the manual is a collection of appendices with various useful information about Hugin.

In the following we will briefly describe each part of the manual.

The Tutorial Part

The tutorial serves two purposes. It is an introduction to the idea of using a causal probabilistic network for representing knowledge, and using the Hugin Inference Engine for efficient application of the knowledge. The tutorial part should give you an idea of what is possible, but it contains no discussion about the theoretical background for CPNs or Hugin. An excellent discussion of using CPNs for representing knowledge is given in [Pea88], along with a comparison between CPNs and other means of representing knowledge. The theoretical background for the Hugin Inference Engine is given in [LS88], and expanded in [JLO90]. An introduction is given in [AOJJ89]. See their bibliographies for a list of literature relevant to Hugin.

The tutorial is also an introduction to Hugin Runtime, an environment for building and testing CPNs. It has been written as a step by step examination of one of the examples built into Hugin Runtime. The tutorial should allow a new user of Hugin Runtime to become familiar with the environment for testing CPNs. The tutorial is far from complete, it only mentions the features necessary to go through the example. In particular, no comments on the facilities for building your own CPNs are made. A complete description of Hugin Runtime can be found in the reference part. The tutorial can be used with all versions of Hugin Runtime, even the free Hugin Demonstration version.

The Reference Part

The reference part contains a short but complete description of Hugin Runtime. If you are new to Hugin Runtime, you should probably read the tutorial part first. The tutorial covers everything you need to know in order to test CPNs, but does leave out some of the more seldom used or advanced stuff, and does not cover building your own CPNs. The information in the reference part closely resembles the information available through the context sensitive on-line help.

The Hugin Runtime development environment is divided into a number of windows. The reference manual is organized around these windows.

The CPN Browser (Chapter 10)
First you must choose what CPN to work with. This is done through the CPN Browser. The CPN Browser allows you to choose a directory to work with, and to choose a CPN from that directory. The CPN Browser also supports other operations, such as creating and removing CPNs.

The Base Window (Chapter 9)
After you have chosen a CPN, it will be displayed in the Base Window. From the Base Window, you can both edit and test the CPN. The CPN is displayed graphically in a pane as a collection of nodes, representing domain variables, and directed links representing causal relationships. The Base Window Pane allows you to manipulate these directly. All other windows can be displayed from the Base Window.

The Monitor Windows (Chapter 11)
In order to test a CPN, you will need to enter evidence into the nodes, and to view the probabilities of each state. You can do this by opening monitor windows for each node you are interested in examining.

The CPN and Node Properties Windows (Chapter 12 and chapter 13)
The CPN Properties Window allows you to change the global properties of the CPN, while the Node Properties Window allows you to change the properties of the individual node, such as the probability table, and the names of the states.

The Learning Window (Chapter 14)
The Learning Window provides you with functions needed for updating the potential tables in a CPN by examining new evidence.

The Junction Tree Window (Chapter 15) The Junction Tree Window allows you to view the internal representation of the CPN, and detect conflicting evidence.

The Cases Browser Window (Chapter 16)
The Case Browser allows you to save a set of findings (a *case*) to be reused later.

The Printer Window (Chapter 17)
The Printer Window allows several ways of printing and even previewing the CPN currently displayed in the Base Window.

The Likelihood Window (Chapter 18)
The Likelihood Window allows you to enter relative likelihoods for the individual states of node.

The Tool Properties (Chapter 19)
You can customize Hugin Runtime from the Tool Properties Window. Most of these properties are used for controlling the visual presentation of the CPNs and the monitor windows. Other properties control the compilation of CPNs from editable format to runable format.

The Message Log (Chapter 20)
Error messages are displayed temporarily in the left footer of the appropriate window, and permanently in the Message Log, together with a more detailed explanation.

Help (Chapter 20)
If you press HELP (on some keyboards you should press the F1 key instead) a Help Window appears with an explanation of the item below the mouse pointer.

There are a few other rarely used windows, also described in this chapter.

The Miscellaneous Appendices

Hugin Runtime is intended to follow the OPEN LOOK GUI (Graphical User Interface) specification, and we have used standard OPEN LOOK terminology throughout this manual. The reference part of this manual assumes basic knowledge of OPEN LOOK applications and terminology, and makes no attempt to explain generic OPEN LOOK facilities. However, appendix D contains a short introduction to the OPEN LOOK terminology and facilities, as they are used by Hugin Runtime. Another OPEN LOOK dependency in the manual is to the OPEN LOOK Window Manager. If you are using another window manager, you should also read appendix E.

Hugin Runtime comes with a number of example CPNs (see appendix A). The editable CPNs are stored in a textual format. The format is described in appendix B. Some of the other file formats used by Hugin is briefly discussed in appendix C.

If you have any problems using Hugin Runtime, or if you have any other questions or comments, please contact us at the following address:

Hugin Expert A/S
North Jutland Science Park
Niels Jernes Vej 10
PO Box 8201
DK-9220 Aalborg
Denmark

Phone: +45 98 15 66 44
Fax: +45 98 15 85 50
Email: `info@hugin.dk`

Part I

Tutorial

1

Starting Hugin Runtime

Before we can do anything useful with Hugin Runtime, we must start the application and choose a CPN to work with.

1.1 Before You Start

Since the example described in this tutorial is built into Hugin Runtime, no special installation is required. However, in order to make the context sensitive help work properly, you will need to install the file '`Hugin.info`' at a standard place like '`/usr/lib/help`' or '`/usr/openwin/lib/help`' or, alternatively, define the environment variable '`HUGINHOME`' to point to the directory containing the '`Hugin.info`' file. You can do this from the C shell by typing:

```
setenv HUGINHOME /usr/local/Hugin
```

at the shell prompt, assuming that the '`Hugin.info`' file is installed in '`/usr/local/Hugin`'. Hopefully, these details have been taken care of by the person installing Hugin Runtime at your site.

Hugin Runtime is a *graphical* development environment, and it therefore requires that you are running the X11 Window System. Hugin Runtime has been tested under OpenWindows from Sun and X11R4 (and X11R5) from MIT. You need to know how to start the window system and create at least a terminal emulation window, and how to move and close windows. If you do not know how to do this, the simplest approach is to ask someone that does. In general, you will have to have specific items in your `PATH`, have specific setup files, etc. If you are going to use OpenWindows, OpenWindows User's Guide [Sun90] provides an excellent introduction.

The examples in this tutorial assumes you are using the OPEN LOOK Window Manager, which is the default window manager under OpenWindows, and also part of the X11R4 distribution from MIT. Hugin Runtime has also been tested under the Motif Window Manager and the Tab Window Manager. The differences between using Hugin Runtime under the three window managers has been summarized in appendix E.

In this tutorial, we have used the OPEN LOOK [Sun89] terminology for describing the user interface. In particular, we refer to the mouse buttons by their logical names, see figure 1.1.

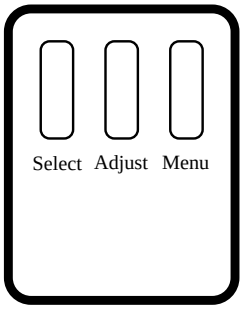


Figure 1.1. *The mouse buttons*

The logical names of the mouse buttons are:

SELECT Usually the left mouse button. Used for selecting objects and activating controls. For example, you select a node by clicking (press and release) SELECT while the mouse pointer is above/inside the node. Similarly, you activate a button by clicking SELECT on it.

ADJUST Usually the middle mouse button. Used for adjusting the selection. If you click ADJUST on an unselected node, the node is added to the selection. Similarly, if you click ADJUST on a selected node, the node is removed from the selection.

MENU Usually the right mouse button. Used for displaying a menu. If you press MENU on a menu button, a menu is displayed. You can now activate an item on the menu by dragging (i.e. moving the mouse without releasing the mouse button) the mouse pointer to an item, and then releasing MENU.

Most operations in Hugin Runtime are performed by selecting some objects and then activating a button or menu item. Other operations are performed by dragging the objects with the SELECT mouse button. See appendix D for more details of how to use the OPEN LOOK GUI.

1.2 Getting Started

It is a good idea to read this tutorial while sitting at a workstation, running the examples given in the manual yourself, thereby getting a feel of how operations are carried out in practice.

When you have started your window system, open a terminal emulation window[1] and enter the command '`xhugin`' at the shell prompt.

If your account has been set up correctly to use Hugin Runtime, two windows will appear. The largest is the empty Base Window illustrated by figure 1.2. If this does not happen, check that the installation has been done correctly.

The Base Window has a border with a title at the top (the header) and some empty space at the bottom. The space at the bottom of the window (called the footer) will be used for displaying error messages and status information, see chapter 7, section 20.2, or section 20.3 for details. The border may look different, if you are not using the OPEN LOOK Window Manager. The various window managers are described in appendix E.

Just below the title is a control area containing four menu buttons and a choice item. Each menu button will display a menu with a number of operations, if you move the mouse pointer to the button and press

[1] A "Command Tool" in OpenWindows or a "xterm" under X11R4.

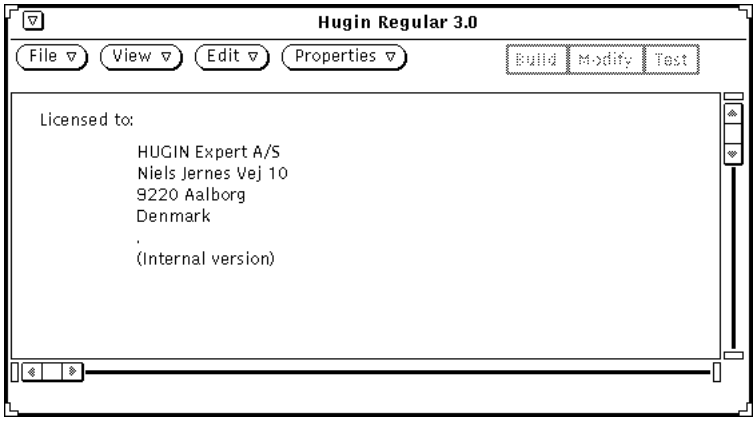


Figure 1.2. *The Empty Base Window*

the MENU mouse button. The choice item can be used for switching between building, modifying, and testing the chosen CPN. Since this tutorial only covers how to test CPNs, we will ignore the choice item. Below the control area, there is a large pane where the CPN eventually will be displayed.

1.3 Choosing a Directory

We now have to choose a CPN to work with. This is done from the second window, The CPN Browser, illustrated by figure 1.3.

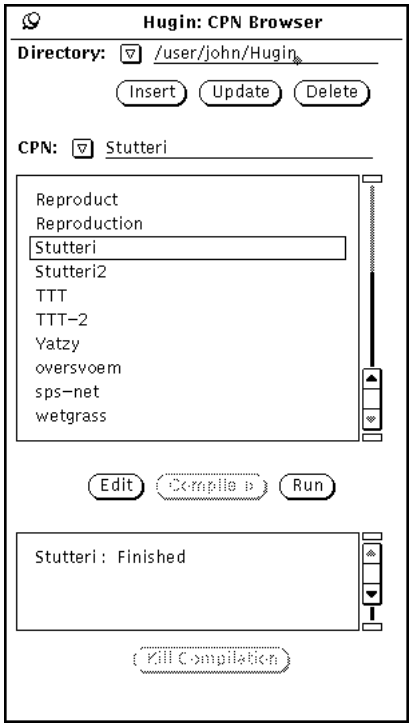


Figure 1.3. *The CPN Browser*

The CPN Browser contains a border, a header, and a footer just like the Base Window. It also contains a number of controls. The first control we need to consider, is the Directory Item at the top of the window.

The Directory Item controls where to search for CPNs. In the example, it will search for CPNs in the directory '`/user/john/Hugin`', which is probably a part of the home directory for a user named 'John'. However, the CPN we want to test is found in a special directory named '`<Demonstration>`', containing the CPNs built into Hugin Runtime.

The Demonstration directory, '`<Demonstration>`', is the default directory in Hugin Demonstration, but in other versions of Hugin Runtime you need to choose the Demonstration directory first. The easy way to do that, is to press the MENU mouse button down while the mouse pointer is above the small triangle next to the '`Directory:`' label. The small button is called an abbreviated menu button in OPEN LOOK, and such buttons always contain a menu. By pressing MENU on the abbreviated menu button, you will display the Directory Menu, as illustrated by figure 1.4.

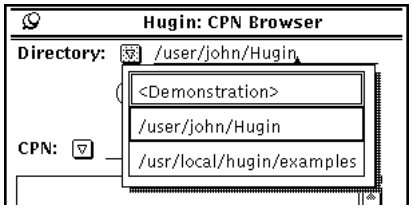


Figure 1.4. *The CPN Directory Menu*

If you drag the mouse to the '`<Demonstration>`' item and release the MENU mouse button, the Demonstration directory will be chosen as illustrated by figure 1.5.

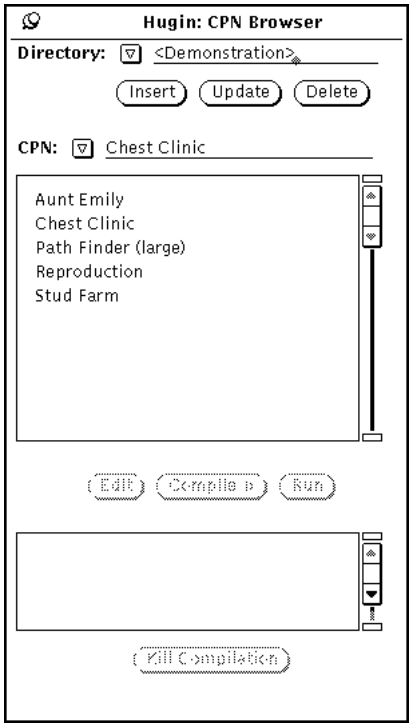


Figure 1.5. *The Demonstration Directory*

If there are no '`<Demonstration>`' item on the menu, you must type it yourself in the Directory Text Field. Click SELECT at the end of the old text ('`/user/john/Hugin`' in the example), remove the old text by pressing `Control U` on the keyboard, and type in the text `<Demonstration>` exactly as written here. After that, activate the Update Button by clicking SELECT on it. You can also insert

the Demonstration directory in the Directory Menu by activating the Insert Button. The Delete Button is used for removing items from the Directory Menu.

1.4 Choosing a CPN

We are now ready to choose a CPN using the next control in the CPN Browser. The CPN Item contains an abbreviated menu button, just like the Directory Item, and a text field where you can type in the name of a CPN. The text field is only used if you will create or rename a CPN, because all known CPNs are already listed in the scrolling list below CPN Item. The abbreviated menu button contains items for creating, renaming, copying, and deleting CPNs. Since we just want to use an existing CPN, we ignore it.

Choosing a CPN is simple. Just click SELECT on the CPN you want to try. We want to try the Chest Clinic example, and we therefore select it as illustrated by figure 1.6.

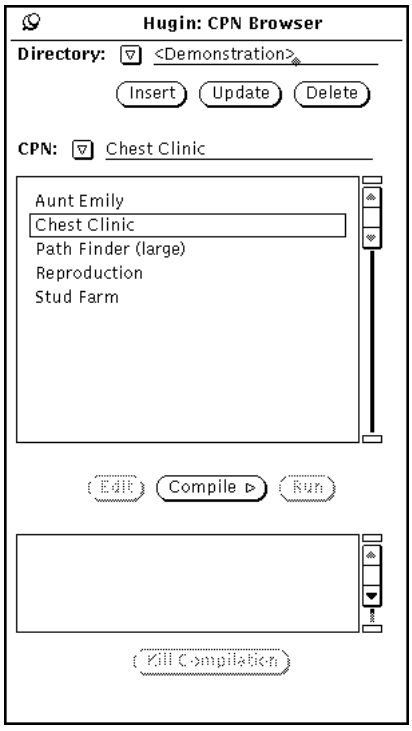


Figure 1.6. *The Chest Clinic is Chosen*

We must now decide whether we want to Edit, Compile, or Run the Chest Clinic CPN. In this case we do not have any choice. As illustrated by figure 1.6, both the Edit Button and the Run Button are dimmed. This is because the built-in CPNs do not exist in a directly editable or runable format. You will need to compile the CPN each time you want to test it. Normally (outside the Demonstration directory) you will always keep the CPN in an editable form, and you will only need to compile the CPN when you have made changes to the editable form. There are no problem in compiling the CPNs from the Demonstration directory, because they are all very small, resulting in very short compile times.

Now activate the Compile Button to load the CPN into the Base Window. The header of the Base Window will be marked as busy while the CPN is being compiled, and then changed to contain the name of the CPN. Figure 1.7 illustrates the new content of the Base Window.

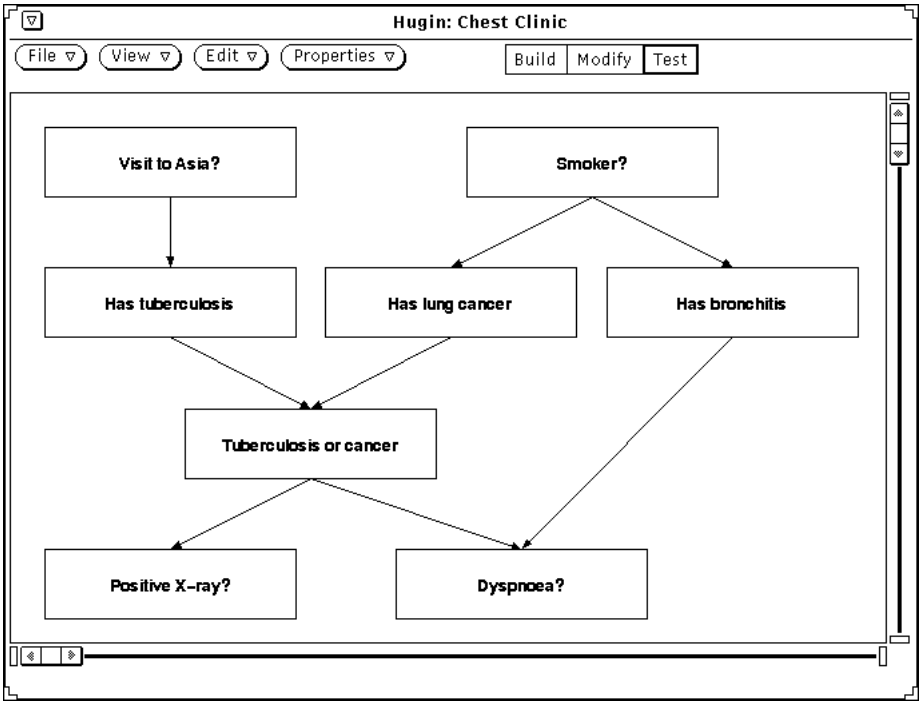


Figure 1.7. *The Chest Clinic example has been loaded*

You may now dismiss the CPN Browser by clicking SELECT on the pushpin at the upper left corner, we will not need it again in this tutorial.

You have now completed the first phase of the tutorial. You have started Hugin Runtime and loaded a CPN. Before we start experimenting with the CPN, we will provide a description of the chest clinic example we just have loaded.

2

The Chest Clinic

The "Chest Clinic" example is based on a small causal probabilistic network. The knowledge embedded in the example is as follows:

> Shortness-of-breath (dyspnoea) may be due to tuberculosis, lung cancer or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The result of a single chest X-ray does not discriminate between lung cancer and tuberculosis, nor does the presence or absence of dyspnoea.

The structure of this knowledge can be represented by a directed acyclic graph, as illustrated by figure 2.1.

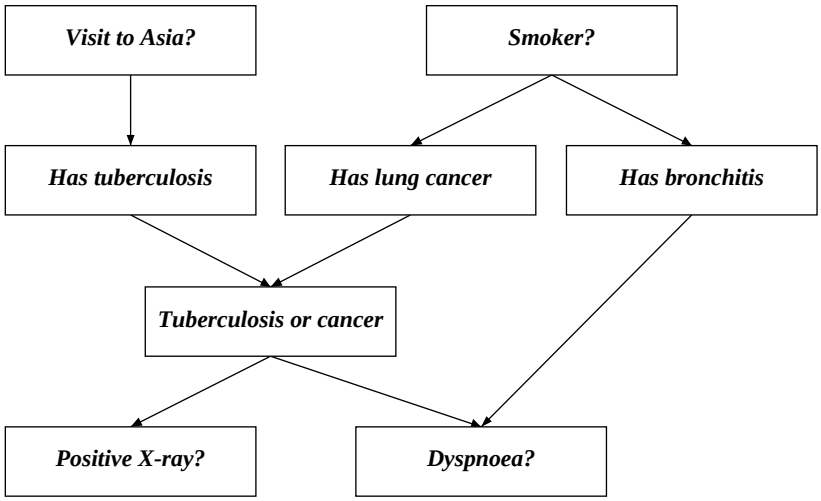


Figure 2.1. *A CPN for the Chest Clinic*

The nodes in the graph represent (random) variables, and the links represent causal (uncertain) relations. The node in the graph labeled "Visit to Asia?" represents the (random) variable holding the knowledge of a patient's possible visit to Asia. The link from "Visit to Asia?" to "Has tuberculosis" represents

the fact that a causal relationship exists between these two variables, i.e. that the probability of a patient having tuberculosis is affected by the patient having been or not been to Asia in the past.

Each node in the graph has a finite set of states. In the Chest Clinic example, each node has the states "yes" and "no". The variable, which the node represents, is always in exactly one state. However, our knowledge about the state of the node is incomplete, and this incompleteness is represented by probability distributions on the state sets of the node. This means that a probability is assigned to each state of the node, representing our current *belief* in the node being in that state.

The causal relations in the knowledge base are represented as conditional probabilities. In the Chest Clinic example, smoking is a risk factor for lung cancer, as represented by the link from the "Smoker?" node to the "Has lung cancer" node in figure 2.1. The probability for the "Has lung cancer" node being in either state is therefore dependent on the state of the "Smoker?" node. We represent this in a *conditional probability table* as given in figure 2.2.

		Smoker?	
		Yes	No
Lung cancer	Yes	0.10	0.01
	No	0.90	0.99

Figure 2.2. *The conditional probability table for lung cancer given smoking. Note: The table reflects a population of patients at the chest clinic rather than the entire population.*

In the table, the probability for the node "Has Lung Cancer?" being in the state "Yes" is 0.10 if "Smoker?" is in the state "Yes" and 0.01 if "Smoker?" is in the state "No". In other words, there is a 10% chance of a patient having lung cancer if that patient is a smoker, whereas the chance (or risk) of a non-smoker having cancer is only 1%.

As can be seen from the CPN, nodes may have a varying number of parents. A node like "Smoker?" has no parents. This means that a patient being a smoker is not caused by any other variable in our model. The states for such a node are assigned *a priori* probabilities. This means that we assign each state a probability that is obtained from somewhere outside the model. In this case we can use simple statistics to learn that half of the patients that enter our lung clinic are smokers, that is, the states "Yes" and "No" both get a probability of 0.50.

Other nodes have more that one parent. The node "Dyspnoea?" (shortness of breath), for example, has two parents, "Tuberculosis or Cancer" and "Has Bronchitis". This means that both of these factors can cause a patient to experience shortness of breath. In the model this is reflected by the conditional probabilities for the states of the node "Dyspnoea?" being dependent on the states of *both* of the other nodes.

Dyspnoea?	(yes, no)	Tuberculosis or cancer	
		Yes	No
Has bronchitis	Yes	(0.90, 0.10)	(0.70, 0.30)
	No	(0.80, 0.20)	(0.10, 0.90)

Figure 2.3. *The conditional probability table for dyspnoea (shortness of breath) given "Tuberculosis or Cancer" and "Has Bronchitis". The vectors in the table represent '(Yes, No)' for "Dyspnoea".*

The probability table for the node "Dyspnoea?" therefore becomes a three-dimensional table. To present this table we use probability *vectors* for each combination of parent states (see figure 2.3). In the table we see that if the patient has both bronchitis and either tuberculosis or cancer (or both), the chance of having shortness of breath is 90% (upper left corner of the table), whereas if the patient has bronchitis but neither tuberculosis nor cancer, the probability of shortness of breath is 80% (upper right corner).

In general each node in the graph is given a conditional probability table specifying the probability distribution for its states for each possible combination of the states of its causal parents. In the case of orphans (nodes with no parents) the table reduces to a probability vector for the states of the nodes.

When a knowledge base has been described in terms of causal relations and conditional probability tables, the Hugin system can *compile* this into a structure that can be used directly for reasoning. When the CPN has been compiled, all nodes will have *initial probabilities* reflecting our belief in each state of the node.

3

Creating Monitor Windows

Now, if we return to the CPN you compiled a few pages back (see page 7), you should now learn how to examine the beliefs for the different nodes. You will have to open monitor windows using the Show Monitor Item from the View Menu, see figure 3.1.

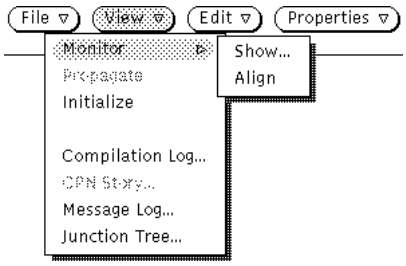


Figure 3.1. *The Show Monitor Item*

First, you must select the nodes you are interested in. There are several ways of doing that. First, you can select a node simply by clicking SELECT while the mouse pointer is on the node. Try to select the node “Smoker?” this way. When a node is selected, it is marked with an extra thick border, as illustrated by figure 3.2.

You can extend the selection by clicking ADJUST on other nodes. Try experimenting a little with the SELECT and ADJUST mouse buttons. Now click SELECT on the “Smoker?” node again. That will select the “Smoker?” node again, while deselecting all other nodes. Now activate the Show Monitor Item from the View Menu. You do that by pressing the MENU mouse button on the View Button at the top of Base Window, dragging the mouse down to the Monitor Item, dragging it right until a new menu appears, and then releasing MENU on the ‘Show” item. This is illustrated by figure 3.1.

After activating the Show Monitor Item from the View Menu a *monitor window* will be displayed as illustrated by figure 3.3.

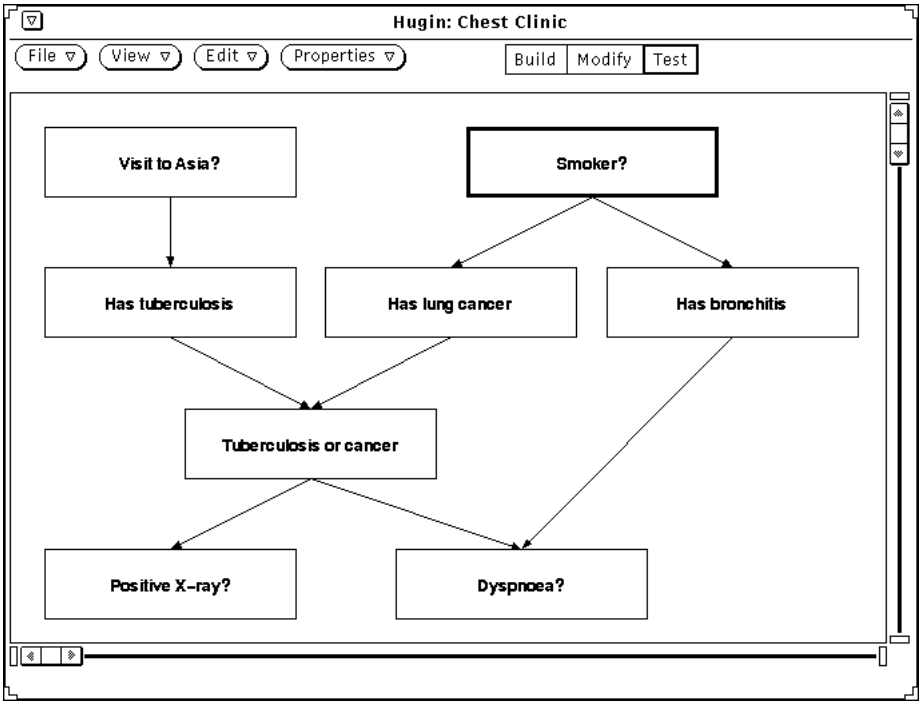


Figure 3.2. *The "Smoker?" node is selected.*

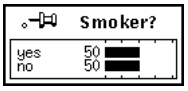


Figure 3.3. *The "Smoker?" Monitor Window*

The Show Monitor Item will create a monitor window for each selected node. The monitor windows will be placed above the corresponding nodes in the Base Window.

The monitor window displays the current belief in each state of the node. Currently it displays the initial probabilities for the "Smoker?" node. The monitor window is just an ordinary window, which you can move or dismiss as usual. To dismiss, just click SELECT on the pushpin in the upper left corner of the window.

The initial beliefs of a node that is not an orphan are computed using ithe conditional probability tables and the initial probabilities of the orphan nodes. The initial beliefs therefore represent the information we can obtain from our model without having extra evidence at our disposal.

We now want to create monitor windows for the other nodes in the CPN. The easiest way to do that is to first activate the Select All Item from the Edit Menu, as illustrated by figure 3.4.

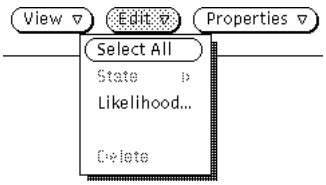


Figure 3.4. *The Select All Item*

If you do not want to be troubled with the View Menu and the Edit Menu, there exist a short cut for activating the two operations. To show all monitor windows, first select all the nodes by clicking SELECT

on the Edit Menu Button, and then display the monitor windows by clicking SELECT on the View Menu Button. When you have displayed all the monitor windows, the Base Window should look as illustrated by figure 3.5.

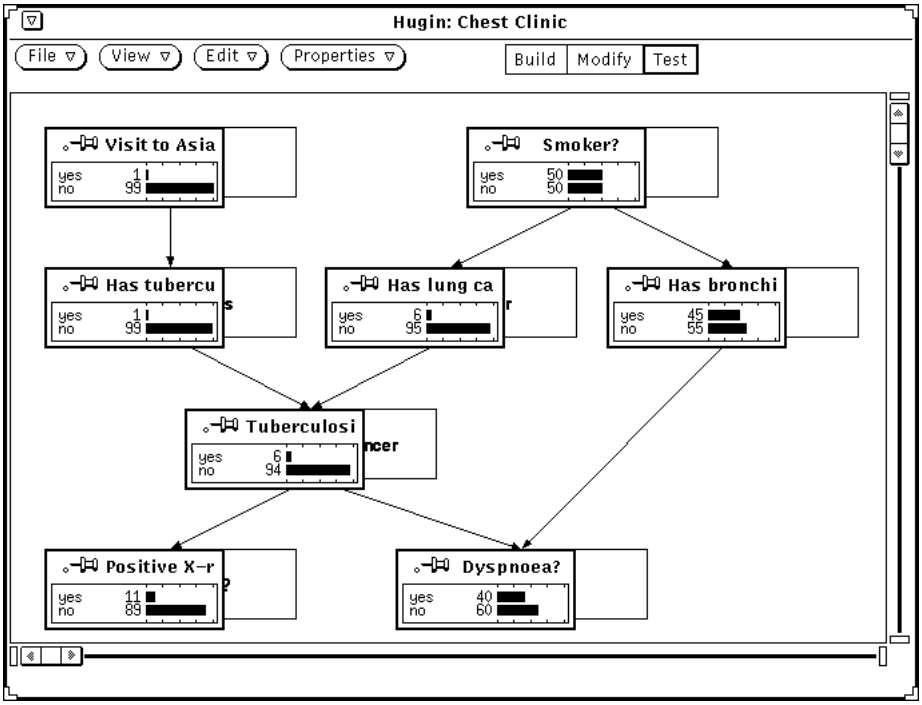


Figure 3.5. *The initial beliefs for the nodes of the Chest Clinic CPN. The probabilities should be interpreted in such a way that out of 100 patients in the lung clinic 40 of them will have shortness of breath and 11 of them will have a positive X-ray test.*

The monitor windows display the initial beliefs in the nodes, thereby telling us what to expect of a new patient arriving at the chest clinic (according to the model). We see that we expect it to be more probable that a new patient suffers from bronchitis (45%) than from tuberculosis (only 1%).

Note that these figures reflect the population of *patients* not the general population. The fact that a person comes to the chest clinic increases our beliefs in that person suffering from a disease. Had the model been for the general population in Denmark, say, we would expect a much lower probability for bronchitis (and the other diseases).

4

Experimentation

4.1 Entering and Propagating Findings

When a new patient arrives at the chest clinic, we will try to make a diagnosis. We could start by asking the patient some questions, such as asking whether the patient is a smoker or not. If the answer is yes, we must enter this new *finding* into the model. Move the mouse pointer inside the monitor window for the "Smoker?" node and click SELECT on the "yes" bar.

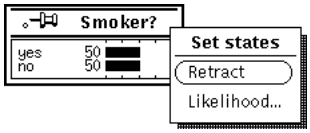


Figure 4.1. *The Set states menu of the "Smoker?" node.*

To retract the finding, you can press the MENU mouse button. This causes a menu titled "Set states" to appear (see figure 4.1). Activating the Retract Item from this menu, will cause all findings for the node to be retracted. You can also toggle an individual state by pressing ADJUST on the appropriate bar in the monitor window.

When a finding is entered, the monitor windows changes to reflect the new situation (see figure 4.2). The monitor window for the node "Smoker?" changes to display that the belief in state "yes" is now 100%. At the same time, the bar showing the belief changes from black to white[1] to indicate that the node has a finding. This makes it easy to quickly recognize the nodes for which findings have been entered.

When a finding is entered the beliefs of the other nodes no longer reflects the latest information. We say that the beliefs are *incoherent* or *out of date* with the findings entered. To reflect this, the bars in the monitor windows change from black to gray. Note that the monitor windows display the same belief values as before the finding was entered. Again, the change in color makes it easy to recognize that the beliefs have not yet been updated.

[1] On a color screen, it will change from green to red.

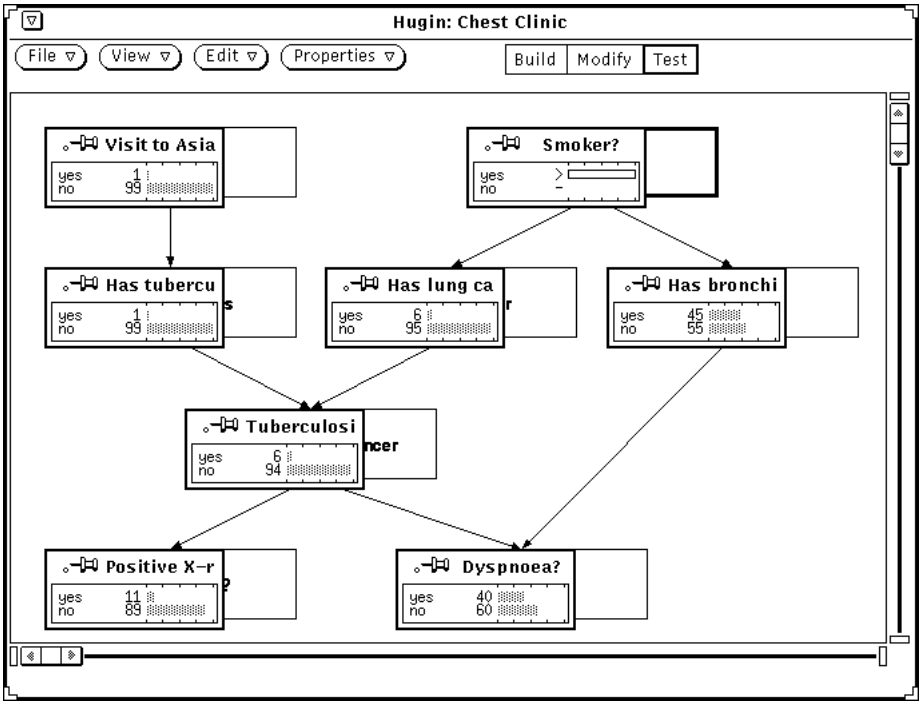


Figure 4.2. *A finding has been entered for node "Smoker?". The monitor window for "Smoker?" displays the finding while the other monitors have their displays grayed out to indicate that the beliefs displayed are incoherent with the findings.*

The View Menu should now look as illustrated on figure 4.3. If you compare it with figure 3.1 you will notice that the Propagate Item is no longer dimmed. This is because new evidence has been entered, which need to be propagated to the rest of the network.

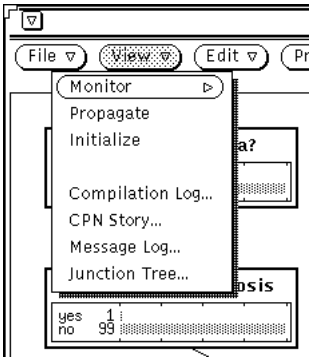


Figure 4.3. *The Propagate Item*

Now activate the Propagate Item in order to make the effects of the finding for the "Smoker?" node to be propagated throughout the CPN, updating all beliefs to reflect the new evidence.

The monitor windows will now be changed to display the new belief values, as illustrated by figure 4.4. The color of the bars in the monitor windows are at the same time changed back to black to reflect that the beliefs displayed are now coherent with the findings.

Looking at the new beliefs we note that the knowledge that the patient is a smoker has increased our belief in the patient suffering from bronchitis (from 45% to 60%). Likewise our belief in the patient

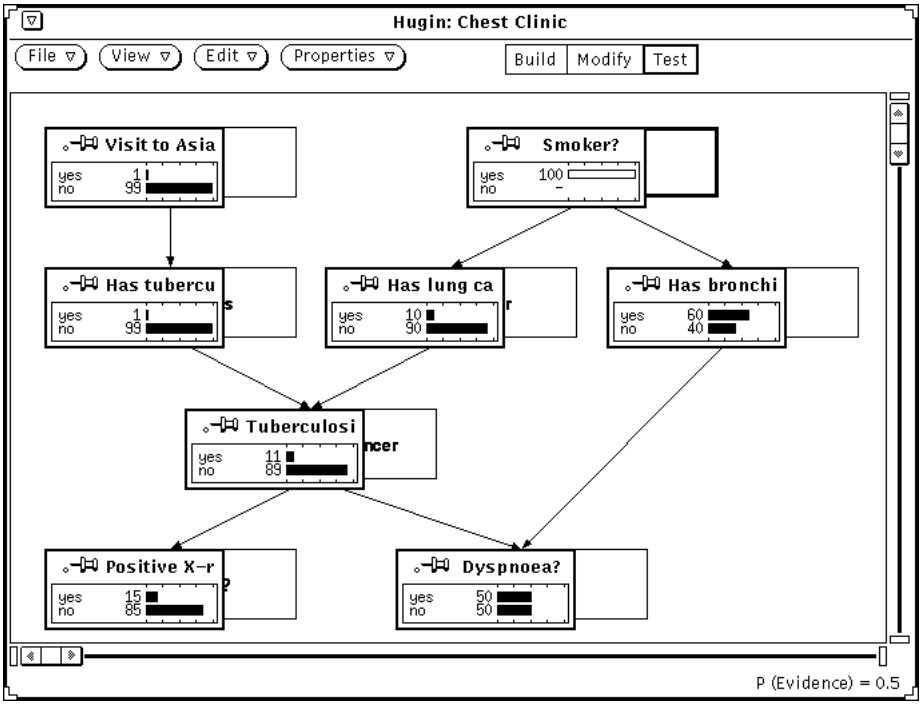


Figure 4.4. *The CPN has been propagated and the monitor windows display the effects of the finding "Smoker?" = "yes".*

suffering from shortness of breath and having a positive X-ray has increased. On the other hand, our belief in the patient suffering from tuberculosis has not changed.

The message

```
P(Evidence) = 0.5
```

is now displayed in the right footer (the bottom right corner) of the Base Window. This indicates that 50% of the total population in the chest clinic matches the findings we have entered, i.e. they are smokers. We already know that, but this information will be useful in more complex examples, as shall be demonstrated later.

From the above example we see that propagation is the central method of reasoning used in the Hugin system. When a model (knowledge base) has been created, all reasoning is performed by entering findings and using propagation to update beliefs to take into account the evidence in the findings.

4.2 Initializing and New Evidence

We have now seen that Hugin can absorb evidence in an orphan node and propagate the effects to the entire CPN. Hugin also allows evidence to be entered into other nodes. To try this, we will first restore the CPN to its initial state. You initialize the CPN by activating the Initialize Item from the View Menu (see figure 4.3). Initializing loads the initial values from the CPN stored on disk, thereby restoring all nodes to their initial beliefs. Note that all monitor windows are updated, and the message "Initialized" is displayed in the right footer.

Now imagine that the patient had arrived at the chest clinic complaining about a shortness of breath. This provides us with a finding that we can immediately enter. Move the mouse pointer to the node

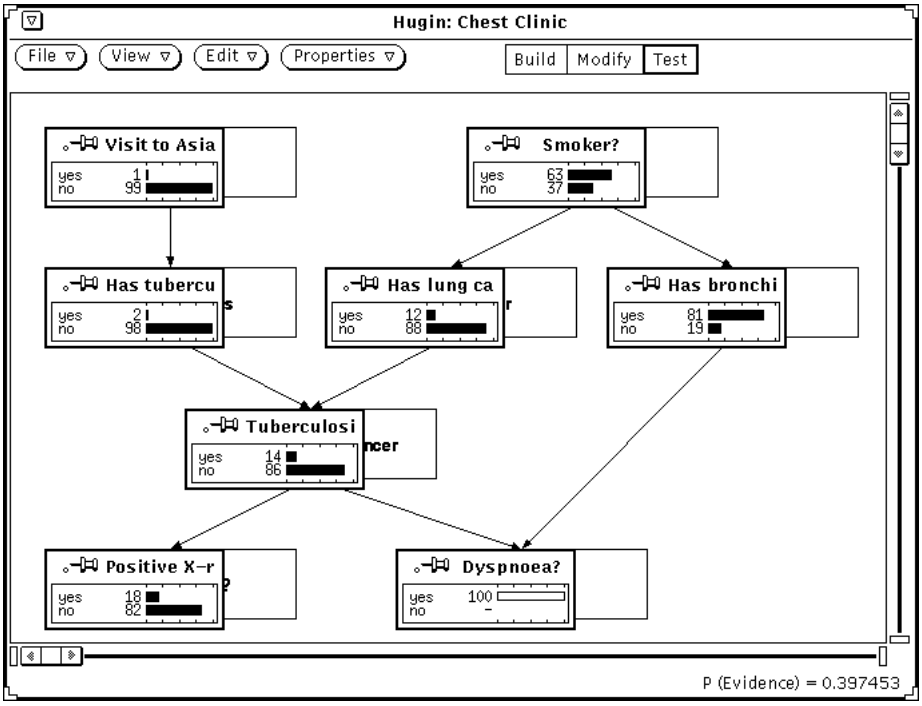


Figure 4.5. *The patient has a shortness of breath. This increases our belief in the patient suffering from either of the diseases.*

"Dyspnoea?" and set the state to "Yes". Note again that the findings are highlighted with a white bar and that the other nodes of the CPN are marked as incoherent with gray bars.

Now activate the Propagate Item from the View Menu in order to update all the nodes, so you can see the effects of the finding that the patient has a shortness of breath (see figure 4.5).

We see that the belief in the patient having tuberculosis, cancer, or bronchitis all increases. Note also that the belief in the patient being a smoker is increased. Smoking is a cause for lung cancer and bronchitis so smoking is indirectly a cause for dyspnoea.

Note that in this example the Hugin Inference Engine reasons "backwards", i.e. from effect to causes. Learning of an effect leads us to expect that the causes are true. The Hugin system will take findings entered in any node of the CPN and during propagation update the beliefs of all other nodes accordingly.

4.3 Simultaneous Findings

To diagnose the patient having shortness of breath we will ask two questions. We will ask if the patient is a smoker and if the patient has recently been to Asia. Suppose the patient answers "no" to the first question and "yes" to the second. To enter this, first move the mouse pointer to the "Visit to Asia?" node and select the "yes" state. Now move the mouse pointer to the "Smoker?" node and select "no". We now have the situation illustrated by figure 4.6 with three nodes displaying a finding and the other nodes being displayed as incoherent.

Activating the Propagate Item from the View Menu at this point will propagate *both* of these findings simultaneously and update the beliefs of all nodes to reflect this, as illustrated by figure 4.7. We see that the belief in the patient having tuberculosis has grown, whereas the belief in the patient having lung cancer or bronchitis has decreased, though the belief in bronchitis is still high.

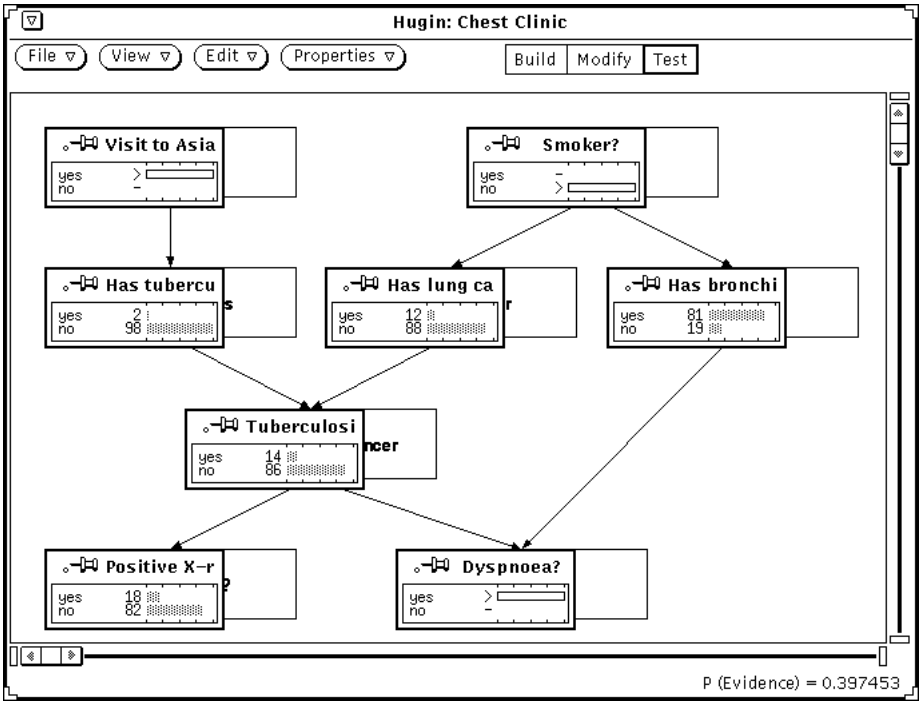


Figure 4.6. *The two findings "Visit to Asia?" = "yes" and "Smoker?" = "no" has been entered but not yet propagated.*

The right footer now displays the message:

```
P(New) = 0.00393411 - P(All) = 0.00156362
```

"`P(All)`" serves the same purpose as "`P(Evidence)`" did before. We expect 0.16% of the chest clinic clients to be non-smokers, who have visited asia, and have shortness of breath. "`P(New)`" indicates the probability of the new evidence entered (namely that the patient does not smoke, and that he has visited asia), given what we already know (namely that he had shortness of breath). That is, we expect 0.39% of the patients with shortness of breath to answer no when we ask them if they smoke, and yes when we ask them if they have visited asia.

The above demonstrates that the Hugin system can propagate any number of simultaneous findings at the time. Moreover, the time required for a propagation is only to a small degree affected by the number of findings being propagated (the effect might both lengthen and shorten the propagation time).

For small networks such as the Chest Clinic example this is not very important, but for large or very large networks where the propagation time can be several seconds or even minutes, the advantage of entering several findings at once before propagating is clear.

To finish the diagnosis, we execute an X-ray test. Suppose this test is positive. We enter this by choosing "yes" from the "Set state" menu of the "Positive X-ray?" node. By propagating we obtain the result illustrated by figure 4.8.

The positive X-ray test increases the belief in the patient suffering from tuberculosis (66%). The belief in cancer has also increased but is still small (13%). The belief in bronchitis has *decreased* and is now lower than the belief in tuberculosis (42%). We are led to conclude that the patient most likely suffers from tuberculosis.

In this last example it is worth noting the decrease in the belief of bronchitis. The finding that the patient has a positive X-ray is not directly related to bronchitis, but the positive X-ray test increases our belief

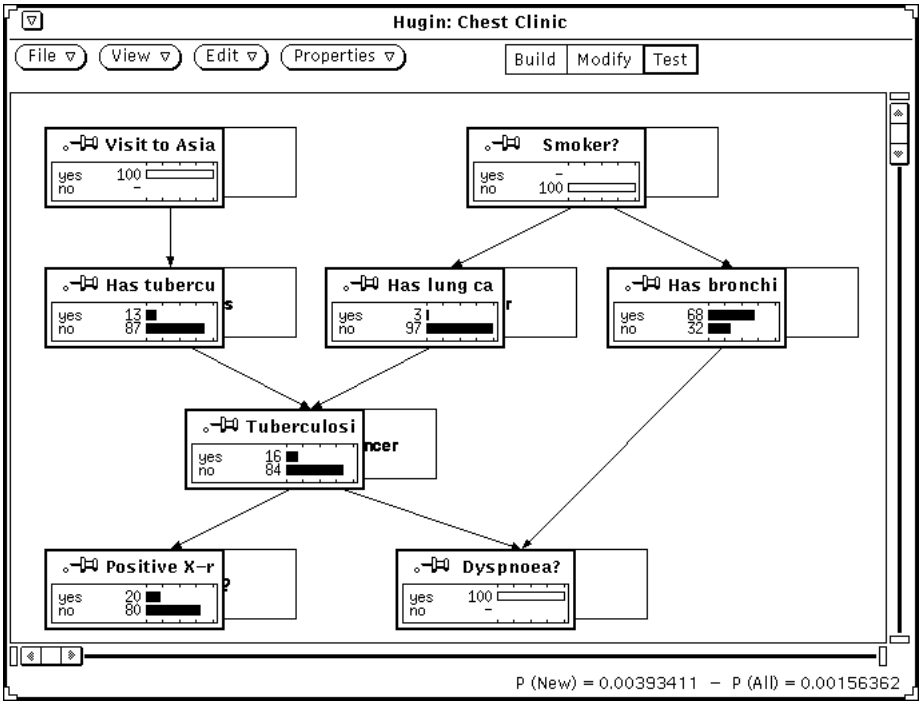


Figure 4.7. *The patient has shortness of breath, has been to Asia but is not a smoker. The probability of bronchitis is high while the probability of tuberculosis has grown but is still moderate.*

in tuberculosis, thereby providing us with an alternative explanation for the patient having shortness of breath. Shortness of breath originally leads us to a high belief in bronchitis, but since we now have another explanation (tuberculosis) that is supported by the finding of a positive X-ray, we are lead to conclude that bronchitis was *not* the cause of dyspnoea.

This effect is called *explaining away* or *non monotonous reasoning* and is an intrinsic feature of causal probabilistic networks. Rule based (expert) systems or systems using fuzzy logic will not handle this example correctly.

4.4 Retracting Findings and Correcting Errors

If you make an error when entering a finding, for example enter the finding into the wrong node, or if you want to see the difference a finding makes, you can use the Retract Item at the bottom of the “Set states” menu to “take back” a finding.

In the above example, suppose you want to know just how important the finding of dyspnoea is for the diagnosis or that the shortness of breath is on closer inspection found to be not very serious. If you move the mouse button to monitor for the “Dyspnoea?” node and activate the Retract Item from the “Set state” menu, you will see that the bar of the monitor changes from white to gray (see figure 4.9).

The belief of the patient having shortness of breath is still displayed as being 100%. This reflects that the retraction has not been propagated and so no new belief values have to be computed. This includes the “Dyspnoea?” node itself. The bars of the other monitors has been grayed out, too. The retraction potentially affects all nodes of the CPN.

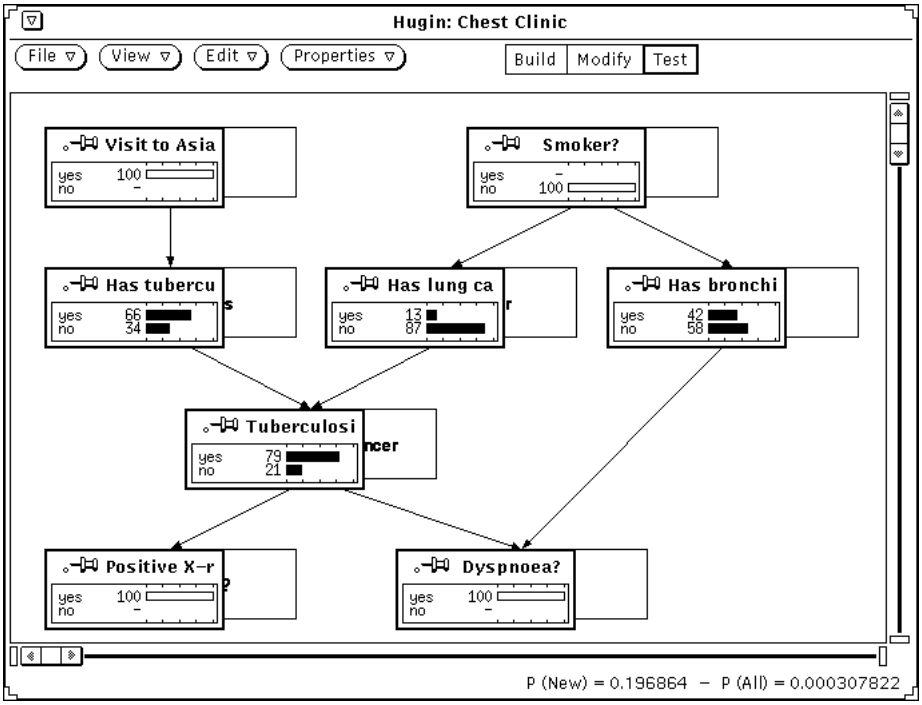


Figure 4.8. *The patient has a positive X-ray test. The belief in tuberculosis is now high while the belief in bronchitis has decreased. The belief in lung cancer has increased but is still small.*

To make the beliefs reflect the new situation a propagation must be performed. Activate the Propagate Item from the View Menu. The result (figure 4.10) shows that the beliefs in all diseases decrease. We now have very little reason to suspect bronchitis (no smoking and no dyspnoea), while tuberculosis is still our best explanation for the positive X-ray.

After retracting the evidence, the right footer only displays "`P(evidence)`" (and a value) after propagation. This is because "`P(new)`" does not make sense unless all the original evidence is preserved.

You are encouraged to try further combinations of evidence for yourself. Experimenting with the Hugin system, repeatedly entering and retracting findings, propagating, and observing the changes in the beliefs is an excellent way of becoming familiar with the system.

Experimentation is also a good way to get a feel for how causal probabilistic networks operate and are used. Try, for example, to change the finding of the node "Smoker?" from "no" to "yes" to see the effect of smoking. How does this affect the diagnosis? Note that you do not have to retract the finding first, the new finding can be entered directly in place of the old one.

You are encouraged also to experiment with the other example CPNs provided with the Hugin Runtime system. These examples are the case of aunt Emily (a murder case), the reproduction example (analysis of an inheritable disease), and the automobile example (finding faults in the electrical system of an automobile). Please refer to appendix A for a description of the examples.

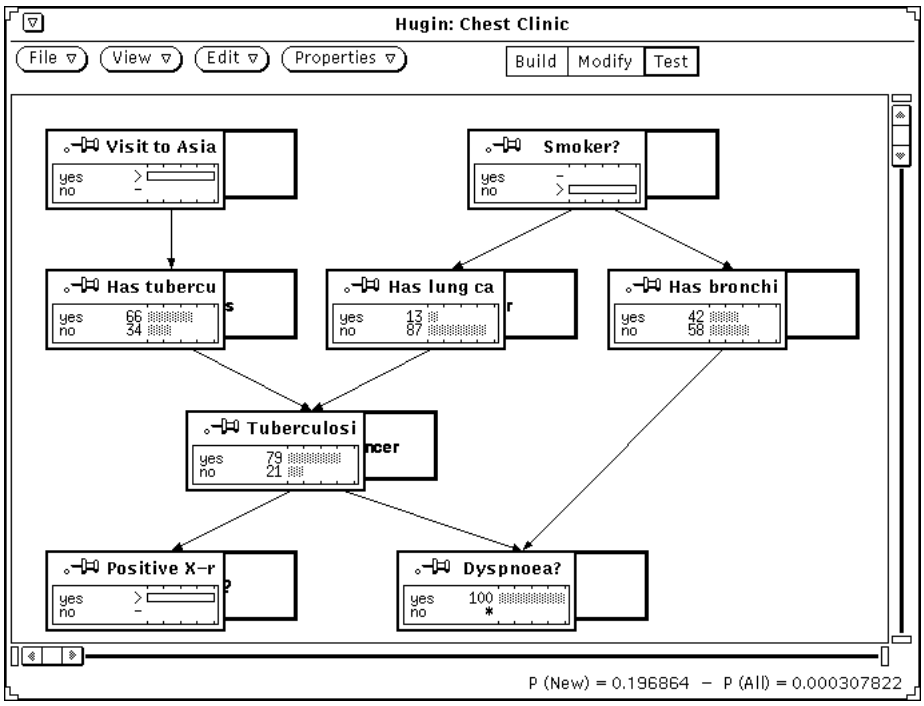


Figure 4.9. *The finding of shortness of breath has been retracted. The beliefs are now incoherent. Note that the belief for dyspnoea is still 100%.*

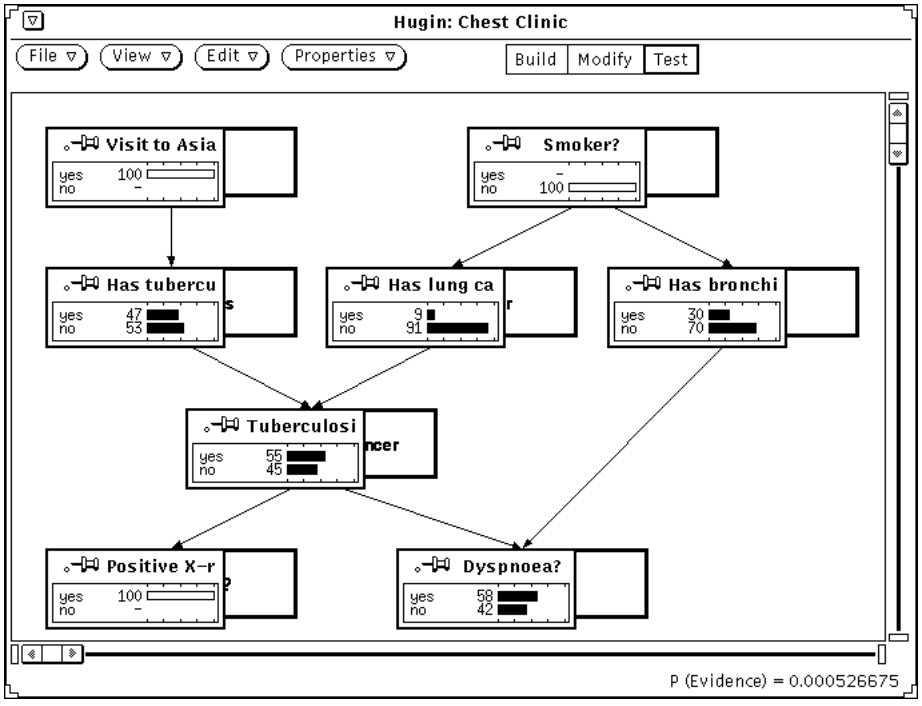


Figure 4.10. *The retraction of shortness of breath has been propagated. The beliefs in all three diseases has decreased. Tuberculosis still dominates.*

5

A Note on CPN Development

When developing a knowledge base the process of repeatedly entering findings, initializing, trying new findings, retracting, and so on, is often a very useful way of understanding the properties of the CPN under construction. The fact that evidence can be entered into any node of the CPN and the effects observed makes it possible to use the Hugin Runtime system as a powerful knowledge base "debugger".

During development it is often useful to do *predictive* reasoning, e.g. to enter a conclusion and see if the expected premises are found. It is often easier for a human to make judgments in the "opposite direction" of the way the system will be used in normal operations, i.e. it is often easier for a domain expert to predict the symptoms of a specific failure than to predict the failures that are causing a specific symptom.

As an example, it is hard to know how strongly a positive X-ray should affect our belief in a patient having cancer, as there could be other causes for the X-ray being positive, and establishing the correct weights is difficult. On the other hand we are pretty sure that *if* the patient has lung cancer the X-ray test will be positive. To see if our CPN reflects this, we can initialize the CPN and enter the single finding "Has lung cancer" = "yes" and propagate. This gives the result illustrated by figure 5.1.

As we would expect, the X-ray test of a lung cancer patient should be positive. We also see that we expect most patients suffering from lung cancer will be smokers and having a shortness of breath. Interestingly, many patients suffering from lung cancer will also suffer from bronchitis.

Although not covered by this tutorial, Hugin Runtime includes a complete environment for building CPNs. Since the CPN editor is an integrated part of Hugin Runtime, a single mouse click is enough to switch between testing and building a CPN. Read the reference part of the manual to learn more about how to build and edit CPNs from inside Hugin Runtime.

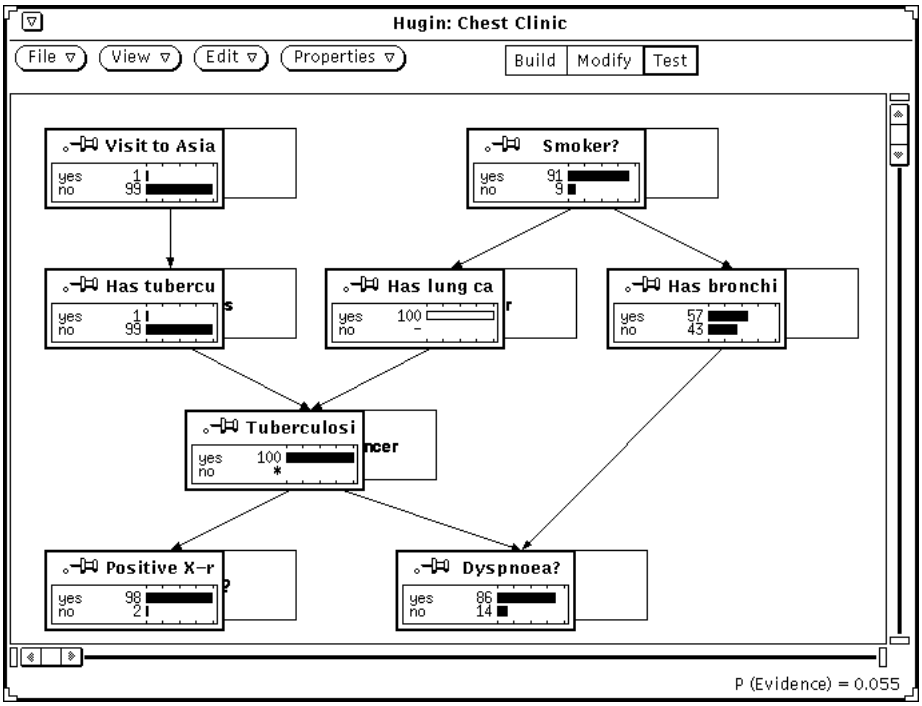


Figure 5.1. *Predictive cancer. If the patient has lung cancer, the X-ray test should be positive. The patient is probably a smoker.*

6

The On-line Help System

Hugin Runtime includes a very extensive context sensitive help system, covering almost all the material from the reference part of this manual. You simply position the mouse pointer somewhere on one of the Hugin windows, and press the HELP key on the keyboard. For example, move the the mouse pointer to the View Menu Button in the Base Window, and press HELP. A help window should now appear, as illustrated by figure 6.1.

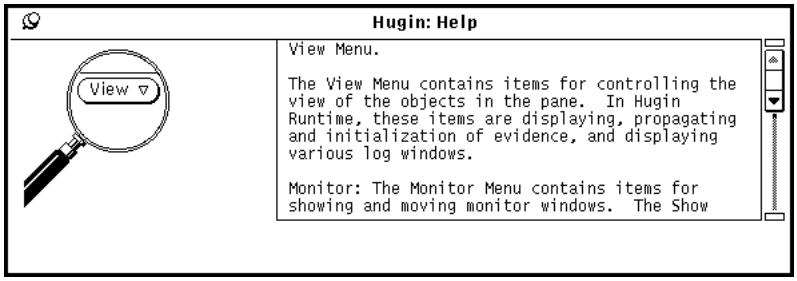

Hugin: Help
View
View Menu.
The View Menu contains items for controlling the view of the objects in the pane. In Hugin Runtime, these items are displaying, propagating and initialization of evidence, and displaying various log windows.
Monitor: The Monitor Menu contains items for showing and moving monitor windows. The Show

Figure 6.1. *The Help Window*

If your account is not set up properly to use Hugin, one of two problems may arise:

- Nothing happens when you press HELP. Try using the F1 key instead. This is also necessary if you are using a keyboard without a HELP key. If that does not help, try to bind the F1 key to HELP by executing the command

  ```
  xmodmap -e 'keysym F1 = Help'
  ```

  from a terminal emulator, such as 'xterm' or 'Command Tool'. You can make the binding permanent by inserting the command in the '`.xinitrc`' file in your home directory.

- A notice labeled 'No help available for Hugin:view' appears. This is a symptom that the file '`Hugin.info`' has not been installed properly. The file should be in one of the following directories:

`/usr/lib/help`

The default directory for help files.

`$OPENWINHOME/help`

The directory with help files for OpenWindows. `$OPENWINHOME` is an environment variable pointing to the directory where OpenWindows is installed. That is usually in the directory `/usr/openwin`.

`$HUGINHOME`

The directory where Hugin has been installed. You should set the `HUGINHOME` environment variable to point to the directory containing `Hugin.info`.

`.`

(dot) The current directory. You can always start Hugin Runtime from the directory containing `Hugin.info`.

7

Error Handling

Error messages will be shown in the left footer of the appropriate window, as illustrated by figure 7.1.

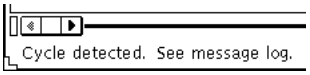


Figure 7.1. *The Left Footer*

The appropriate window is the window where you activated the operation which caused an error. The error message will disappear as soon as you click with the mouse or type a key in any window controlled by Hugin Runtime.

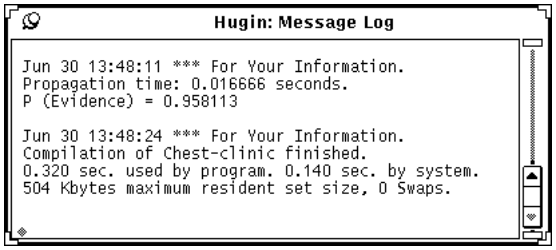

Hugin: Message Log
Jun 30 13:48:11 *** For Your Information.
Propagation time: 0.016666 seconds.
P (Evidence) = 0.958113
Jun 30 13:48:24 *** For Your Information.
Compilation of Chest-clinic finished.
0.320 sec. used by program. 0.140 sec. by system.
504 Kbytes maximum resident set size, 0 Swaps.

Figure 7.2. *The Message Log Window*

The error messages are also displayed in the Message Log Window, accessible from the View Menu in the Base Window (see figure 4.3).

The Message Log Window, as illustrated by figure 7.2, contains a log of all error messages and warnings, often including a more detailed explanation than would fit the left footer.

Other more or less useful information are occasionally also shown in the Message Log Window. This includes messages from the background compilation of CPNs, propagation time messages, conflict measures in the junction tree etc.

8

Leaving Hugin

When you have finished experimenting with Hugin Runtime, you can leave the system by activating the Quit Item from the File Menu, as illustrated by figure 8.1.

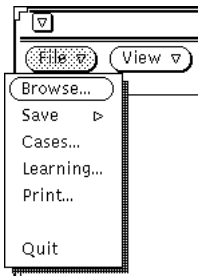


Figure 8.1. *The Quit Item*

You can leave Hugin Runtime by other means, depending on the window manager you are using, but activating the Quit Item from the File Menu will always work.

Note, that leaving the system destroys all the findings you have entered.

You can restart the system at any time to do further experiments, but each time you load a CPN it will start up initialized.

Part II

Reference Manual

9

The Base Window

The Base Window (see figure 9.1) contains a pane with a graphical representation of a CPN. The CPN can be manipulated directly from the pane, or through a number of controls in the control area above the pane. These controls consist of four menu buttons at the left side of the window, and a mode setting at the right side of the window.

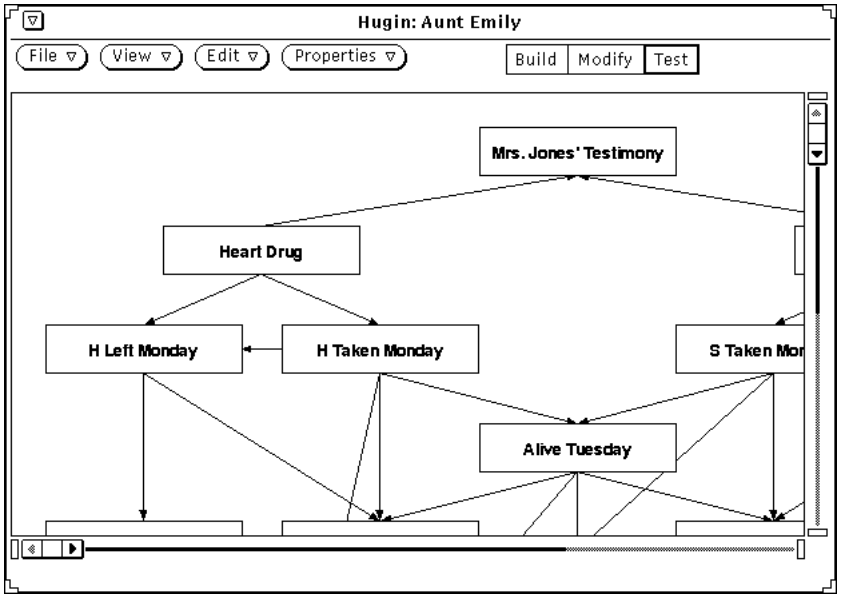


Figure 9.1. *The Base Window*

The very first thing you want to do, is to load a CPN. You do that by choosing Browse from the File Menu. See chapter 10 for a description of how to load a CPN from the CPN Browser.

9.1 The Base Window Pane

The Base Window Pane contains a graphical representation of a CPN. The graphical representation can be partly controlled from the Tool Properties Window, see section 19.4. As well as giving a graphical presentation of the CPN, Hugin Runtime allows you to manipulate the CPN through the Base Window Pane. From within the Base Window Pane you can select objects and then perform operations on the selected items by activating various menu items, or you can move or create objects by direct manipulation.

9.1.1 Selecting

Selected nodes are emphasized with an extra thick border, as illustrated by figure 9.2. Similarly, selected links are emphasized with an extra thick line.

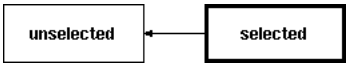


Figure 9.2. *Selected and Unselected Nodes*

You can select a node by clicking SELECT[1] on the node in either Modify Mode or Test Mode. In Modify Mode you can also select links by clicking SELECT on them. If you click SELECT on an object, you deselect all other objects. If you click SELECT outside any object, all objects are deselected.

You can extend the number of selected object by clicking ADJUST on an unselected selected or unselected objects. This will select the object, without deselecting any other object. Similarly, clicking ADJUST on an selected object deselects it, without affecting other objects. You can use ADJUST in all modes, including Build Mode. Links can only be selected in Modify Mode and Build Mode.

You can select all objects by activating the Select All Item from the Edit Menu (see section 9.5) or the pane's pop-up menu.

9.1.2 Operations

In Modify Mode and Build Mode, you can delete the selected nodes and links by activating the Delete Item from the Edit Menu (see section 9.5) or the pane's pop-up menu. You can also edit the properties of the selected node, by activating the Node Item from the Properties Menu (see section 9.6) or from the pane's pop-up menu.

In Test Mode you can display monitor windows for the selected nodes by activating the Show Monitor Item from either the View Menu (see section 9.4) or the pane's pop-up menu.

9.1.3 Direct Manipulation

In Modify Mode you can move a node by pressing SELECT on the node, dragging the node to the new position, and then releasing SELECT. When you drag the node, a dimmed outline of the node is shown, as illustrated figure 9.3.

In Build Mode you can create a node by clicking SELECT at the position where you want the node created. You can also create a link from one node to another by pressing SELECT on the first node, dragging the pointer to the other node, and then releasing SELECT.

[1] For more details about how to handle the mouse, see appendix D.

Figure 9.3. *Moving a Node*

9.1.4 The Pane's Pop-up Menu

The pop-up menu (see figure 9.4) in the pane provides fast access to the most frequently used items from the menus in the control area. All the menu items can also be activated from the four menu buttons at the top of the window, and will be described together with these four menus. Some of the items in the menu are dimmed in order to illustrate that they cannot currently be activated.

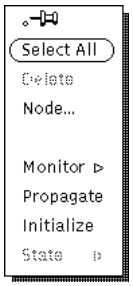


Figure 9.4. *The Pane's Pop-up Menu*

9.2 The Mode Setting

The pane can be in one of three modes of operation. You choose the mode by clicking SELECT on the Mode Setting item illustrated by figure 9.5.

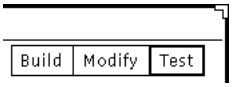


Figure 9.5. *The Mode Setting*

The modes are optimized for the following phases:

Build Mode Building the structure of the CPN by creating nodes and links between nodes.

Modify Mode Modifying the properties of the nodes by selecting the nodes and displaying their property windows.

Test Mode Testing the CPN by displaying monitor windows, entering evidence, and propagating the evidence.

The modes control what happens when you click SELECT in the pane, or when you drag the pointer in the pane (You drag the pointer by pressing SELECT and move the pointer without releasing SELECT). The modes also control which operations (from the menus in the control area or the pane) are available.

9.2.1 Click SELECT

In Modify Mode and Test Mode you can select nodes by clicking SELECT on the node. In Modify Mode you can also select links by clicking SELECT on them. In Build Mode, you create a node by clicking SELECT in the pane.

9.2.2 Dragging

In Modify Mode you move a node by pressing SELECT on the node, dragging the node to the new position, and then releasing SELECT. In Build Mode you create a link from one node to another, by pressing SELECT on the first node, dragging the pointer to the other node, and then releasing SELECT. Dragging has no effect in Test Mode.

9.2.3 Operations

In Modify Mode and Build Mode you can delete nodes and links, display CPN and node property windows, and save the CPN. In Test Mode you can display monitor windows, enter evidence, and propagate the evidence.

All other operations and mouse buttons are independent of the mode.

9.3 The File Menu

The File Menu (see figure 9.6) contains items affecting things outside the application itself. In Hugin Runtime, this consists of browsing, saving, case reading and writing, learning, printing, and quitting.

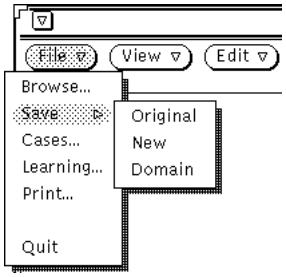


Figure 9.6. *The File Menu*

Browse Display the CPN Browser, from which you can create or load a CPN. See chapter 10.

Save Save the CPN currently shown in the pane. You can either save the CPN under the original name, or under a new name taken from the CPN Browser's Directory Item (see section 10.1) and CPN Text Field (see section 10.2).

Original Save the CPN currently shown in the pane with the same name, and in the same directory, as it was originally loaded from. Save Original is disabled if the CPN has not been changed. If you want to save the CPN anyway, use the Save New item.

New Save the CPN shown in the pane under the name currently shown in CPN Browser's CPN Text Field. It is possible to copy the CPN across directories using the Directory Item in the CPN Browser. Save New is always enabled.

Domain Save the CPN including all findings you may have entered, and all changes you may have made to the potential tables (either directly or through the learning features). See also chapter 16. Hugin will propagate first if necessary. Save Domain is only enabled in Test Mode.

Cases Display the Case Browser, where you can load and save sets of findings. See chapter 16.

Learning Display the Learning Window, where you can let the CPN learn from specific cases. See chapter 14.

Print Display the Printer Window, where you can print the current CPN. See chapter 17.

Quit Leave the application.

9.4 The View Menu

The View Menu (see figure 9.7) contains items for controlling the view of the objects in the pane. In Hugin Runtime, these items are displaying, propagating and initialization of evidence, and displaying various log windows.

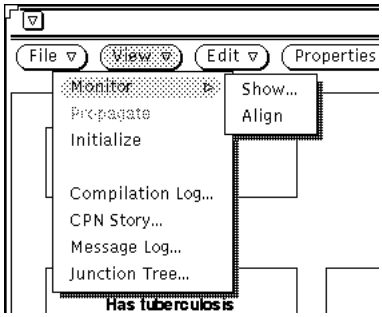


Figure 9.7. *The View Menu*

Monitor The Monitor Menu contains items for showing and moving monitor windows. Monitor windows are described in chapter 11. The Monitor Menu is only enabled in Test Mode.

Show Display a monitor window for each selected node. The monitor windows allows you to examine the probability of each state in the node, as well as entering and retracting evidence for the node.

Align Move all displayed monitor windows to the position in the pane of the node they are representing. This is useful after moving the Base Window. Monitor windows with the pushpin in will not be aligned.

Propagate Propagate all the findings entered since the last propagation or initialization. Propagation allows you to see how the findings affect the probability of the states in each node. The Propagate Item is only enabled in Test Mode, and then only if the CPN needs propagation. See also chapter 11.

Initialize Retract all findings, and display the a priori values for all nodes. This item is only enabled in Test Mode.

Compilation Log Display the Compilation Log Window. It contains information about the internal structures created during compilation. The Compilation Log is described in section 20.4.

CPN Story Display the CPN Story Window. This window contains an explanation of the CPN currently displayed in the Base Window Pane. Only the built-in CPNs from the Demonstration Directory have an associated story. The CPN Story Window is described in section 20.5.

Message Log Display The Message Log Window. The Message Log Window contains a log of all error messages and warnings, often including a detailed explanation. Other more or less useful information is occasionally also shown in the Message Log Window. The Message Log is described in section 20.2.

Junction Tree Display the Junction Tree Window. The Junction Tree Window contains a graphical view of the internal data structures used by Hugin. The Junction Tree is described in chapter 15.

9.5 The Edit Menu

The Edit Menu (see figure 9.8) contains items for manipulating the CPN shown in the pane. In Hugin Runtime, this consist of selecting and deleting. Other manipulations are performed directly in the pane.

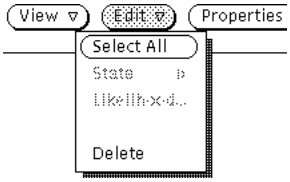


Figure 9.8. *The Edit Menu*

Select All Select all nodes in the current CPN. In Modify or Build Mode, also select all links.

State Display a menu of all possible states for the selected node, from which you can choose one or none. This is used for entering findings to the network. This is not implemented yet. Currently, you have to set the state through the Monitor Windows.

Likelihood Display the Likelihood Window. You can enter likelihood for the individual states from the Likelihood Window. See chapter 18 for details.

Delete Delete all selected nodes and links. This item is disabled in Test Mode. In order to delete nodes or links, the Base Window must be in Modify Mode. You should then select the nodes and links by clicking SELECT or ADJUST on them, before you activate the delete operation.

9.6 The Properties Menu

The Properties Menu (see figure 9.9) contains items for displaying property windows. The property windows allow you to change the properties of a node, a CPN, or the application itself.

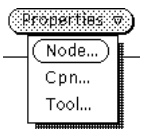


Figure 9.9. *The Properties Menu*

Node Display the Node Properties Window (chapter 13). You can set the properties for the selected node from the Node Properties Window. These properties include the name and label of the node, the number and names of the states, and the probability table associated with the node. This item is enabled when exactly one node is selected.

CPN Display the CPN Properties Window (chapter 12). You can set the properties of the CPN currently being shown in the pane from this window. These properties include the name and label of the CPN, the size of the nodes, and the current scale factor. This item is enabled when the Base Window is in Modify Mode or Build Mode.

Tool Display the Tool Properties Window (chapter 19). You can set properties affecting more than a single CPN or node from this window. Most of these properties are used for controlling the visual presentation of CPNs and Monitor Windows. Other properties control the compilation of an editable CPN to a runable CPN.

10

The CPN Browser

The CPN Browser (see figure 10.1) contains controls for selecting a directory, for selecting a CPN, and for editing, compiling, or running the selected CPN in the Base Window.

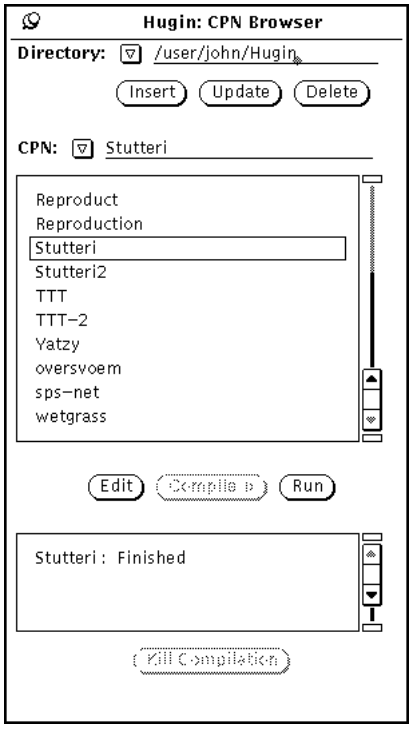


Figure 10.1. *The CPN Browser*

10.1 The Directory Item

The Directory Item allows you to choose the directory where Hugin Runtime will search for CPNs. You can either choose a directory from the abbreviated menu button (see figure 10.2), or you can write the

name of the directory directly in the text field.

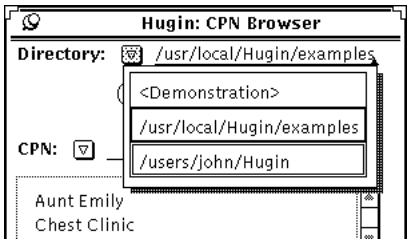


Figure 10.2. *The Directory Menu*

If you edit the text field directly, you must either press RETURN in the text field, or click SELECT on the Update Button, in order to update the CPN list (i.e. reading the contents from disk).

You can customize the Directory Menu with the Insert Button and the Delete Button. Click SELECT on the Insert Button to insert the directory shown in the Directory Text Field to the Directory Menu, or click SELECT on the Delete Button to delete the directory shown in the Directory Text Field from the Directory Menu.

You can make the changes to the Directory Menu permanent by clicking SELECT on the Set Default Button in the Tool Properties Window, refer to section 19.5.

10.2 The CPN Item

The CPN Item has three components: A list of CPNs, a text field, and an abbreviated menu (see figure 10.3) with a number of items for manipulating the CPN List.

You can select a CPN from the CPN List by clicking SELECT on the CPN. This operation will copy the name of the selected CPN to CPN Text Field. You can now edit the CPN Text Field and press RETURN in order to rename the CPN.

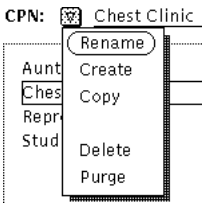


Figure 10.3. *The CPN abbreviated menu*

A number of operations are available from the CPN Menu. All the operations, except create, use the selected CPN. Some operations also use the content of the CPN Text Field.

Rename Rename the selected CPN to the name in the CPN Text Field.

Create Create an empty CPN with the name in the CPN Text Field.

Copy Copy the selected CPN to a new CPN with the name in the CPN Text Field.

Delete Delete the selected CPN (this is *not* reversible!).

Purge Remove the compiled version of the CPN. This is useful in order to save disk space, or if you want to force a recompilation.

10.3 The CPN Buttons

When you want to load the selected CPN into the Base Window Pane, you must activate one of the buttons below the list of CPNs.

Edit Click SELECT on the Edit Button in order to edit the selected CPN. The CPN will be in Modify Mode, once loaded to the Base Window. The Edit Button will be dimmed if the no CPN is selected, or if the selected CPN does not exist in an editable form.

Compile Click MENU on the Compile Button to display the compilation menu, illustrated by figure 10.4.

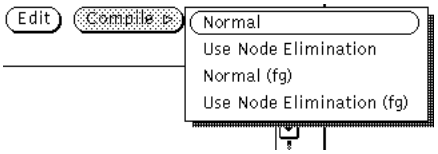


Figure 10.4. *The Compile Menu*

The CPN may be compiled in the background, in which case compilation progress is displayed in the scrolling list below the Compile Button, or it may be compiled in the foreground, in which case progress is displayed by a gauge replacing the Edit, Compile, and Run buttons.

Background compilation allows you to continue using Hugin Runtime while the compilation takes place, but foreground compilation uses less system resources, and automatically displays the CPN in Test Mode in the Base Window Pane when done. You can abort a background compilation job by selecting it in the compilation list, and activating the Kill Compilation Button.

Do *not* attempt to compile a CPN in the background while using it in Test Mode. Hugin Runtime reads the CPN from disk during normal use in Test Mode. If the CPN is changed on disk while using it, anything might happen.

Normal compilation uses the algorithm chosen in the Compilation Category in the Tool Properties Window (see section 19.2) to find the node elimination sequence. This node elimination sequence is saved, so you later can compile with the same node elimination sequence. Reusing the node elimination sequence will only work if you have not made structural changes to the CPN (i.e. adding, removing, or renaming nodes). The node elimination sequence can be modified manually, see section C.1.

The Compile Button will be dimmed if the no CPN is selected, or if the selected CPN does not exist in an editable form, or if the selected CPN has not been changed since last time it was compiled. To force a recompilation, remove the runable version by choosing the Purge Item from the CPN Menu.

Run Click SELECT on the Run Button in order to run the selected CPN in the Base Window. The CPN will be in Test Mode, once loaded to the Base Window. The Run Button will be dimmed if the no CPN is selected, or if the selected CPN does not exist in a runable form. You can create a runable form by clicking SELECT on the Compile Button.

11

The Monitor Windows

You can display *monitor windows* for any number of nodes by first selecting the nodes in the Base Window Pane (which must be in Test Mode) and then activating the Show Monitor Item from the View Menu. Each Monitor Window contains information about the current state of a node. Figure 11.1 illustrates a typical Monitor Window.

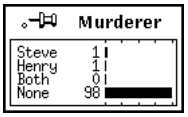


Figure 11.1. *A Monitor Window*

The window's title bar shows the name of the node currently being monitored. Each state of the node is displayed in one row of the window. The rows contain three columns.

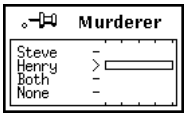


Figure 11.2. *A Monitor Window Containing Evidence*

- First column contains the name of the state.
- The second column may contain:

  a number indicating probability per cent of the state, as calculated from the evidence entered to the system. A '0' indicates a small probability, less than the current precision gives room to express. You can change the precision used from the Monitor Category in the Tool Properties Window (see section 19.3).

  > indicating that a finding has been entered to this state, but not propagated.

  - indicating that a finding has been entered to other states in the node, giving this state a value of exactly 0.

* indicating the probability has been calculated to exactly 0.

\+ indicating that a finding has been entered to this state, but the probability has been calculated to exactly 0 anyway. This is possible for multistate findings.

- The third column contains a bar representing the same probability as described above. There are three different types of bars.

  Black Representing probability that has been calculated according to all the evidence entered to the CPN. The Monitor Window illustrated by figure 11.1 contains bars of this type.

  White Representing evidence. Figure 11.2 illustrates the same Monitor Window after entering evidence.

  Grey Representing probability calculated before the latest evidence was entered. Figure 11.3 illustrates a Monitor Window which is inconsistent with the evidence. Activate the Propagate Item from the View Menu in the Base Window (see section 9.4) to make all Monitor Windows consistent with the evidence.

  On a color screen white will be replaced with red and black with green, unless you have chosen different colors from the Tool Properties Window.

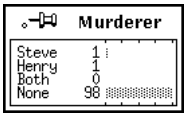


Figure 11.3. *An Inconsistent Monitor Window*

It is possible to adjust the size of each field, as well as the precision used by the second field, from the Tool Properties Window (see section 19.3).

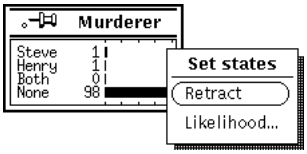


Figure 11.4. *The Monitor Menu*

You can enter a single state finding by clicking SELECT on a state bar. Doing that will retract the findings for all other states. You can adjust a finding by clicking ADJUST on a state bar, which will toggle the state between found and not found without affecting the other states, thus making multistate findings possible.

You can retract a finding or enter likelihood by pressing MENU in the Monitor Window, and activating one of the items. The menu is illustrated by figure 11.4.

12

The CPN Properties

You can set the properties of the CPN currently being shown in the Base Window Pane from the CPN Properties Window, illustrated by figure 12.1. You can display the CPN Properties Window by activating the CPN Item from the Properties Menu in Modify Mode or Build Mode. The CPN properties include the name and label of the CPN, the size of the nodes, and the current scale factor.

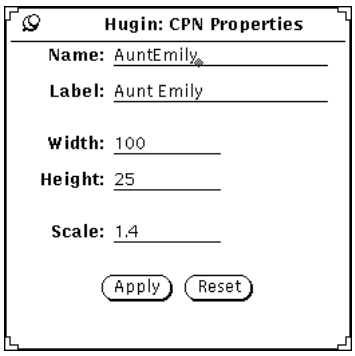


Figure 12.1. *The CPN Properties Window*

12.1 The Property Items

12.1.1 Name

The Hugin API allows you to have multiple CPNs active simultaneously. In order to distinguish between the CPNs, each must have a unique name. Hugin Runtime does not currently make any use of the name.

The name can only contain letters, digits, and the special characters '_', '-', '/', and '.'. The name is not allowed to start with a digit.

12.1.2 Label

The Hugin API allows you to attach a label to each CPN, which can be used by an application as a more descriptive name for the CPN.

Although it is considered a bad style, multiple CPNs may have the same label. The label can contain any characters except double quotes ‘"’. Space ‘ ’ is a valid character in a label.

12.1.3 Width

This item controls the width of the nodes, measured in points. One point is 1/72 inch.

12.1.4 Height

This item controls the height of the nodes, measured in points. One point is 1/72 inch.

12.1.5 Scale

You can set the scale factor to change the size of the CPN display. The scale factor affects both the size of the nodes and the distance between the nodes. If you double the scale factor, the CPN will take four times as much area on the screen to display[1].

Setting the scale factor can be used as a primitive zoom facility.

12.2 The Apply Button

Click SELECT on the Apply Button to apply the current settings to the CPN.

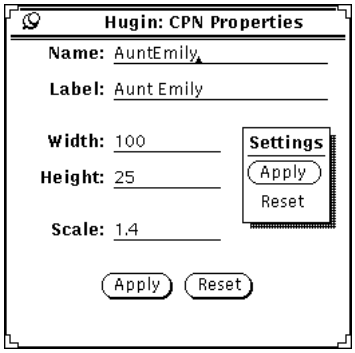


Figure 12.2. *The Settings Menu*

12.3 The Reset Button

Click SELECT on the Reset Button to reset the content of the CPN Properties Window from the CPN.

[1] As you well know: 2 * 2 = 4 !!

12.4 The Settings Menu

It is also possible to Apply or Reset the CPN Properties from the Settings Menu (see figure 12.2), which is available by pressing MENU in the background area of the CPN Properties Window.

13

The Node Properties

You can set the properties for the selected node from the Node Properties Window, illustrated by figure 13.1. You display the Node Properties Window by activating the Node Item from the Properties Menu. You must have exactly one selected node. The node properties include the name and label of the node, possible parents of the node, the number and names of the states in the node, and the probability table associated with the node.

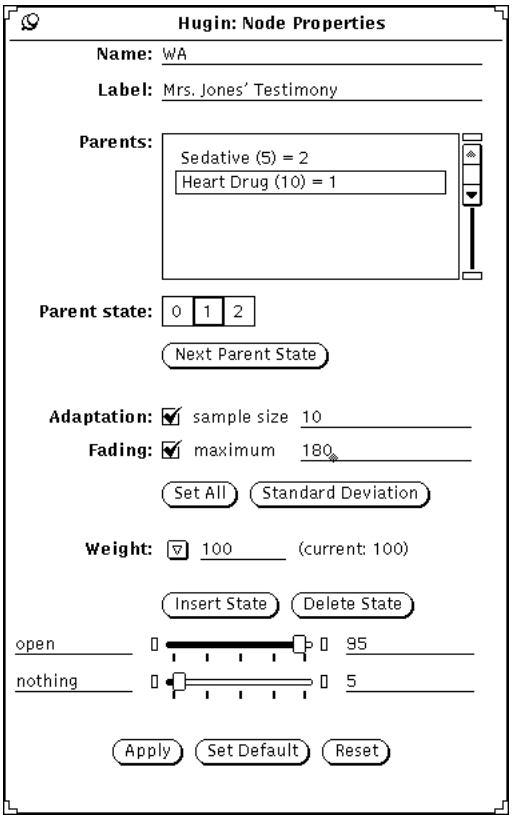


Figure 13.1. *The Node Properties Window*

Most node properties can only be changed when the CPN is in Build or Modify Mode, but the probability table may also be changed in Test Mode.

13.1 Name and Label

13.1.1 Name

Each node must have a unique name in the CPN. The name is of little use in Hugin Runtime, but can be valuable for people who want to access the CPN directly at the API level.

The name can only contain letters, digits, and the special characters '_', '-', '/', and '.'. The name is not allowed to start with a digit.

13.1.2 Label

The node's label is used for identifying the node on the Base Window Pane and the monitor windows.

Although it is considered bad style, multiple nodes may have the same label. The label can contain any characters except double quote '"'. Space ' ' is valid in labels.

13.2 Parent State Combinations

The probability table associated with node contains the relative probability of each state for each possible combination of parent states. In order to easily toggle between the different combinations of parent states, we have three items:

- The Parent Item displays all parents, together with their current state, in a scrolling list. You can select a parent by clicking SELECT in the scrolling list.
- The Parent State Item displays the possible states of the selected parent, either as an exclusive setting, as illustrated by figure 13.1, or an abbreviated menu button, as illustrated by figure 13.2. You can choose a new state for the selected parent by clicking SELECT on the exclusive choice item, or by pressing MENU on the abbreviated menu button.

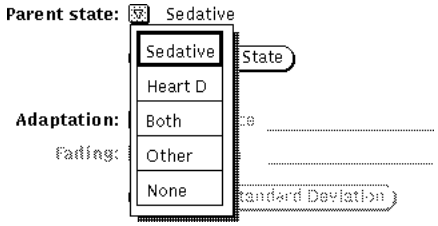


Figure 13.2. *The Abbreviated Parent State Menu*

- The Next Parent State Button provides a fast method for browsing through the different parent state combinations. Activating the Next Parent State Button will cycle through all possible combinations of parent states.

13.3 Adaptation

Adaptation is the learning strategy for Hugin that is used by the Adapt and Learn items in the Learning Window (see chapter 14). The name refers to the fact that this learning strategy allows a running system to adapt to new evidence. Adaptation can be enabled or disabled individually for each parent state configuration, the adaptation items therefore always refer to the parent configuration currently chosen in the Parent Item.

You enable learning by clicking SELECT on the check box next to the Adaptation label. When you have enabled Adaptation, you must enter a sample size, indicating how many cases the current probabilities should be considered equivalent to. This number will be incremented when a new case matching the current parent configuration is adapted. The evidence might be incomplete, that is, findings are missing for some of the parents. In such a case the sample size will still be updated, but with an amount less than 1.0. The sample size may even decrease in extreme cases.

It is possible to specify that old cases should be given less weight than new cases. This is done by checking the box next to the Fading label, and specifying a maximum sample size. This ensures that the sample size does not grow without bounds, but rather stabilizes at the given maximum. This is done by multiplying the sample size by *(m - 1.0) / m* where *m* is the maximum sample size, each time a case has been adapted.

If you want all parent configurations for the node to have the same adaptation parameters, you can do it by specifying the sample size and maximum for one configuration, and activating the Set All Button.

The sample size is one way to specify the uncertainty of the probabilities. Another – often more intuitive – is the standard deviation. Activating the Standard Deviation Button will write the standard deviation for each state calculated from the sample size in the Message Log. The formula used to find the standard deviation is *p * (w - p) / (s + 1.0)*, where *p* is the probability for the state, *w* is the current weight, and *s* is the sample size.

13.4 Weight

By ‘weight’ we mean the sum of the probabilities of each state for the current combination of parent states. It is often custom that the sum should equal 1.0, 100 (per cent), or 1000 (per thousand). The inference engine in Hugin does not care about the sum, it uses only the relative probabilities of the states.

Hugin Runtime remember two different weight values for each combination of parent states: The desired weight and the current weight. The current weight is the sum of the current settings of the sliders, while the desired weight is initialized to the sum of the original settings.

If you change the probabilities using the sliders, the current weight will change as well. If you want to keep the desired (original) weight, you can choose the Normalize Item from the weight menu. You can display the weight menu by pressing MENU on the abbreviated menu button, as illustrated by figure 13.3.

You can also change the desired weight, either by editing the text field directly, or by choosing one of the predefined values from the weight menu. The weight menu also contains a Current Item, which will set the desired weight to the current weight.

Finally, the weight menu contains two special items.

Activating the Uniform Item will force the probabilities for the current combination of parent states to be uniform distributed.

Activating the Reset Item will remove all changes done to the current combination of parent states since last time you activated the Apply Button.

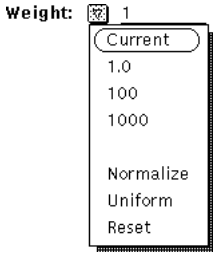


Figure 13.3. *The Weight Menu*

13.5 The Node States

13.5.1 Insert State

Click SELECT on the Insert State Button to insert a new state to node.

Initially, the new state will have probability zero for all combinations of parent states.

13.5.2 Delete State

Click SELECT on the Delete State Button to delete the last state of the node.

Neither the absolute probability nor the relative probability of the other states will be changed for any combination of the parents states.

13.5.3 State Name

The name of each state is written in a text field at the left side of the window. The state name must be unique within a node. The state name is used by Hugin Runtime when entering evidence and when displaying the relative probability for the state.

The state name may contain any characters except double quotes ‘"’.

If you press RETURN you will move to the next state.

13.5.4 State Slider

You can adjust the probability of the state for the current combination of parent states by dragging the slider.

13.5.5 State Probability

The probabilities are written numerically at the right side of the window. You can adjust the probability of the state for the current combination of parent states by entering a new value in the text field.

Press ‘-’ to delete the current value.

Press RETURN to move to the next state.

Pressing RETURN in the last state will move to the next parent state combination, as if you had activated the Next Parent State Button. This provides you with a convenient way of entering data without moving your hand away from the numeric keypad.

13.6 Apply, Set Default, and Reset

In order to make the changes to the Node Properties Window take effect on the node, you must first activate the Apply Button. Activating the Reset Button will reset the settings in the Node Properties Window to the currently active settings. That is, undo all changes made since last time you activated Apply. Please note that if you leave the CPN or Hugin Runtime without activating the Apply Button first, or if you you try to edit the properties of another node, all changes to the current node is lost. You can only edit the properties of one node at a time. Activating the Set Default Button will make the number of states and their names default when you create new nodes. This can be very convenient if you create a CPN with many similar nodes.

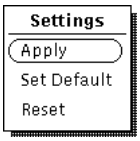


Figure 13.4. *The Node Settings Menu*

It is also possible to Apply, Set Default, or Reset the Node Properties from the Settings Menu (see figure 13.4), which is available by pressing MENU in the background area of the Node Properties Window.

14

The Learning Window

The Learning Window (see figure 14.1) provides you with functions needed for updating the potential tables in a CPN by examining new evidence.

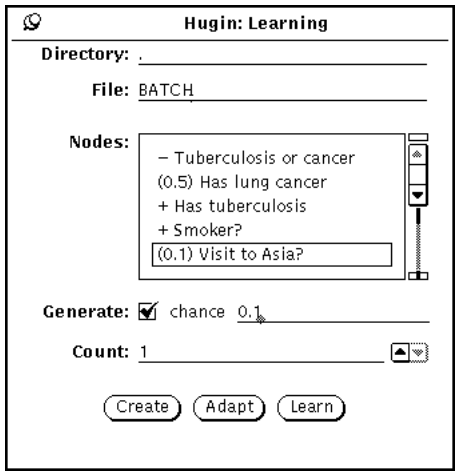


Figure 14.1. *The Learning Window*

The learning window is only active in Test Mode, and will *not* work on compressed CPNs.

You can adapt the CPN in the Base Window to the evidence currently entered into the nodes through the monitor windows or by other means by activating the Adapt Button.

You can learn from a batch file containing a set of cases, which will be used for updating the potential tables in the CPN by activating the Learn Button.

You can create a batch file containing cases generated from the potential tables currently used by the CPN by activating the Create Button.

Before we explain the individual items in detail, lets have a brief explanation about the learning algorithm used.

14.1 Adaptation

The basic idea behind the adaption feature is, that the probability table of a node is interpreted as representing past experience as a contingency table of counts of past cases. Therefore, the initial sample size for each parent configuration has to be set. The larger the sample size, the more sure you are on the probability table, and the more conservative is the adaption to the new cases.

The adaption method has an extra feature, namely *fading*. Technically what happens for a node in fading mode is, that each time a new case is taken into account, the sample size is discounted by a fading factor *q* (less than 1). The result is that the system tends to ignore old cases. Furthermore, in the long run the sample size can not exceed a maximal sample size *1/(1-q)*. Therefore, when a node is declared in fading mode, the *maximum sample size* has to be given.

For a more detailed discussion, please refer to [OLJ92].

14.2 The Batch File

The Learning Window can learn from a batch file or create a batch file. In both cases, it uses the file specified in the Directory and File Text Fields. By default, the directory will be '`.`' indicating the place where Hugin Runtime was launched, and the file name will be '`BATCH`'.

14.3 The Learning List

The Learning List contains the names of all the nodes in the CPN.

If there is a '`+`' symbol in front of the name, findings for this node will be generated if you activate the Create Button. If there is a '`-`' before the name, findings will not be generated. A number before the the name indicates that there is a random chance that findings for this node will be generated.

You can specify whether findings should be generated for a given node, by selecting the node (move the mouse pointer above the node, and click SELECT) and manipulating the Generate Item below the list.

When you activate the Learn Button, the values in the Learning List will be read from the batch file.

14.4 The Generate Item

Set the check box if you want to generate random evidence for the selected node in the batch file. You can specify the chance that evidence for the node are generated to something different from 0.0 or 1.0, by writing a number in the text field.

14.5 The Count Item

Specify how many cases should be generated when the Create Button is activated. When you activate the Learn Button, the Count Item is used for indicating how many cases remains to be processed.

14.6 The Buttons

Create When you activate the Create Button a batch file (specified by the Directory and File Items) with a number (specified by the Count Item) of cases containing randomly evidence for some (specified by the Generate Item) of the node is created.

Adapt Activate this button to adapt the CPN to the finding currently entered.

Learn When you activate the Learn Button, the batch file specified by the Directory and File Items will be read.

The Learning List will be updated to display how many of the nodes have missing data. The Count Item will indicate the progress of the learning process, by showing how many cases remains to be adapted. If you have any monitor windows open, they will indicate the findings as they are entered. The Nodes Properties Window will also update the potentials during the learning process.

A graphical view of the learning process for the first state for the selected parent state combination in the node currently displayed in the Nodes Properties Window can be obtained, by enabling the Show Learning View Toggle in the Miscellaneous Category of the Tool Properties Window.

14.7 The Learning View

The Learning View (see figure 14.2) will display the current value for a single entry in one of the potential tables.

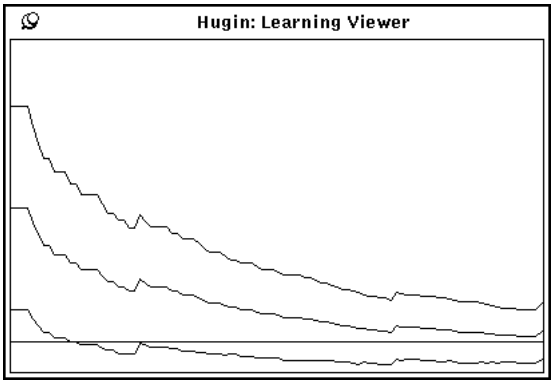


Figure 14.2. *The Learning View*

You select the node by displaying the Node Properties Window (refer to chapter 13) for the node you are interested in, and choosing the parent state combination from that window. The Learning View will always display the value for the first state of the node (for the given parent configuration).

The Learning View will display a single straight line which can set from the Miscellaneous Category of the Tool Properties Window (see section 19.6), and three lines indicating the evolution of the entry while the learning is going on. The middle line represents the potential, while the two other lines twice the standard deviation.

The Learning View is displayed while Hugin Runtime is learning, but only if enabled though the Tool Properties Window.

15

The Junction Tree

The Junction Tree Window allows you to view the internal representation of the CPN, and detect conflicting evidence.

Activating the Junction Tree item from the View Menu in the Base Window will display the Junction Tree Window (see figure 15.1 where the junction tree for the “Aunt Emily” example has been opened). To the left of the window the control panel for manipulating the junction tree is located At the right a pane where the junction tree is drawn. To the right and below the pane two scrollbars are located. The scrollbars can be used to position the view of the junction tree in case it is too big to fit the selected size of the window.

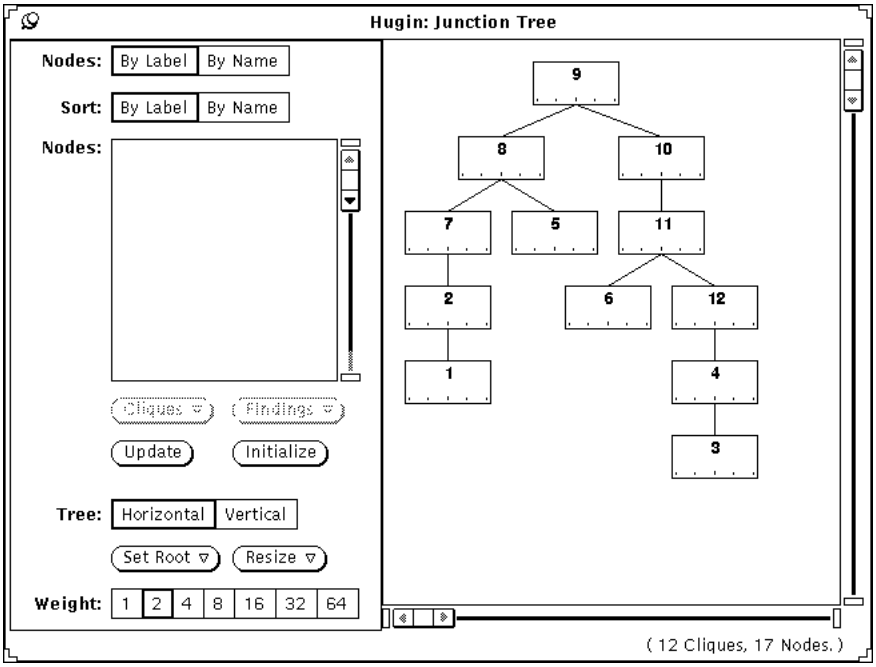


Figure 15.1. *The junction Tree Window. The junction tree for the “Aunt Emily” example is shown. Note that the total number of nodes and cliques are shown in the lower right corner.*

The cliques are numbered from *1* to *n*, where *n* is the total number of cliques in the junction tree. This number is used as the label of the cliques. The numbers are actually the numbering produced during the compilation of the CPN. The clique numbers in the Compilation Log will match the labels (see section 9.4 and section 20.4).

15.1 Layout of the Tree

The default layout of the tree is constructed so the height of the tree will be lowest (the root of the tree is the clique with the shortest longest distance to the leafs). And the default size of the tree is said to be 1.

The layout of the tree can be controlled using the items to the right of the `Tree:` label. (see figure 15.1).

The Horizontal or Vertical Choice controls if the tree should be shown horizontal (the default) or vertical.

Using the Set Root Menu, the root of the tree can be selected. It can either be a selected clique (see section 15.2) or the default layout of the tree. Selecting a new root in the tree will also cause a new propagation from the root of the tree (see section 15.5).

The tree can also be resized. The Resize Menu contains three entries:

Enlarge enlarges the network by 1/4.

Shrink shrinks the network by 1/4. The size of the tree can not be lower than 1/2 (This entry will be deactivated if the size gets 1/2).

Minimal Size sets the size to the smallest, 1/2.

15.2 Selecting Cliques

To select a clique press SELECT inside the clique. The frame of the clique will be wider (blue when using a color display). If one or more cliques are selected and SELECT is pressed in a different clique, the previous selected cliques will be deselected and the new clique selected. Use SELECT to select *just one* clique. To deselect *all* cliques, click SELECT outside the cliques.

To select more than one clique use the ADJUST mouse button. Pressing ADJUST inside a clique will add this clique to the number of selected cliques. An already selected clique can be deselected again by pressing ADJUST on it again.

The MENU button of the mouse is used in the pane to select the Pane Menu (see figure 15.2). In the Pane Menu the first entry, Select All, can be used to select *all* cliques.

Figure 15.2. *The Pane Menu. Activate Select All to select all cliques or Conflicts to print the current conflicts in the Message Log Window (see section 15.5 and section 20.2). The last two entries are similar to the Select Root and Resize menus in the control panel.*

15.3 The Nodes Scrolling List

When one or more cliques have been selected the nodes that are member of these cliques are displayed in the Nodes Scrolling List in the control panel of the Junction Tree Window (see figure 15.3).

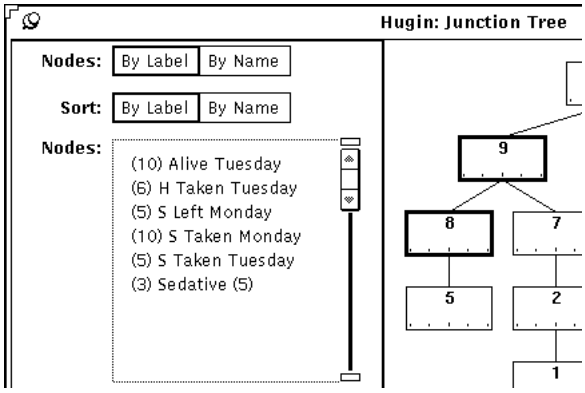


Figure 15.3. *Two cliques have been selected and the nodes that are members of these cliques are shown in the Nodes Scrolling List. The number in parenthesis in front of the labels are the finding clique of the node.*

On top of the Nodes Scrolling List two exclusive choice items control how nodes in the Nodes Scrolling List are displayed. The nodes can be displayed by either label or name, and they can be sorted by either one of those too.

A node can be member of more than one clique, but findings for a node will only be inserted into *one* clique (At present time there is no possibility to enter findings for a node into other cliques than the finding clique). This clique is called the *finding clique* for the node. The names/labels of the nodes in the Nodes Scrolling List are prefixed with the number of the finding clique of the nodes in parenthesis.

15.4 Node Cliques and Findings

When nodes are shown in the Nodes Scrolling List (one or more cliques have been selected) it is possible to get more information on the cliques, to enter and retract findings, and to insert likelihoods into the nodes.

Select a node in the Nodes Scrolling List by pressing Select on the desired node. This way the two menus Cliques and Findings get activated (these menus are deactivated when no node is selected).

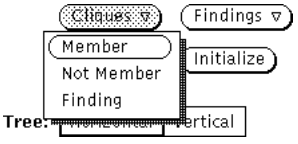


Figure 15.4. *The Clique Menu. Activate the Member item to select the cliques the node is a member of, Not Member to select the cliques where the node is not a member, and Finding to select the finding clique of the node.*

Using the Cliques Menu (see figure 15.4) is is possible to select and deselect cliques. The Cliques Menu has 3 entries:

Member will select the cliques where the node is a member. This will *not* deselect the cliques where the node is not a member.

Not Member will select all the cliques where the node is *not* a member.

Finding will select *one* clique, namely the finding clique of the node.

Findings and likelihood can be inserted or retracted using the Findings Menu (Note again here, that a finding is inserted into the finding clique of the node). When a node is selected in the Nodes Scrolling List the Findings Menu is activated. The menu consists of the state names of the node plus two additional entries: Retract and Likelihood (see figure 15.5).

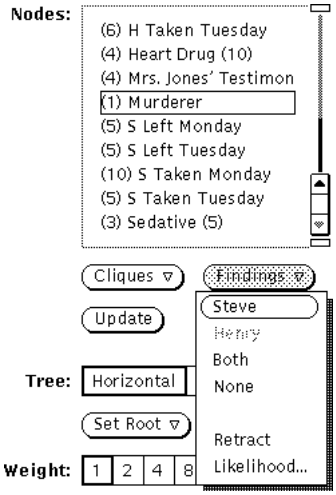


Figure 15.5. *The Finding Menu. Here the node "Murderer" has been selected in the "Nodes" scrolling list. The second state "Henry" have previously been inserted as a finding so it has been deactivated to indicate this. Activating Retract will retract all findings of the node, while Likelihood will bring up the Likelihood Window for the node (see chapter 18).*

Findings are entered by selecting a state name. Only single findings can be entered. To enter multiple findings use the monitor windows.

When a finding has been entered the corresponding state name in the finding menu is deactivated to indicate that a finding has been entered. Note also that only when findings have been entered the Retract Item is activated.

When a finding has been entered a small filled circle (filled with yellow on a color display) will be placed in the clique where the finding has been entered (see figure 15.6).

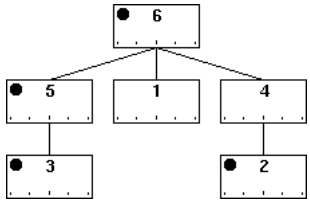


Figure 15.6. *One or several findings have been entered into the nodes 2, 3, 5, and 6, indicated by the circle in the upper left corner.*

15.5 Propagation and Conflicts

In the Junction Tree Window it is possible to calculate conflits. We will not go into details in explaining of conflicts but just give a short definition. Please refer to [JCNJ91] for a detailed definition.

15.5.1 Definition of Conflict

If findings have been entered into some of the nodes in the CPN, say the findings are *a, b, c, ...*, then the conflict measure is equal to:

conf *(a, b, c, ...) = (P(a) * P(b) * P(c) * ...) / P(a, b, c, ...)*

A positive value of *conf* indicates that there may be a conflict between the CPN model and the set of findings. The conflict measure is calculated whenever evidence meet in the junction tree, By using different roots for the special propagation, it may thereby be possible to trace down the origins for a data conflict.

A warning should be given: A high value is only an indication of a possible conflict between data and model. Still, data may be sound and originate from a rare case covered by the model.

15.5.2 Using Conflicts

A special propagation algorithm is used to calculate conflicts during the propagation. This additional propagating algorithm is more time consuming than the normal propagation. Therefore the *only* way to propagate with calculation of conflicts is by pressing the Update Button in the control panel (Update will automatically be "pressed" when a new root of the tree is selected). This button will cause the new propagation to be invoked. It behaves almost the same as the normal propagation except for two important points:

- The propagation is done from the clique that is root of the junction tree.
- The conflicts in the cliques get updated.

The additional propagation algorithm is not available for the special CPNs in the Demonstration Directory (see section 20.5), and the Update Button will therefore be inactive.

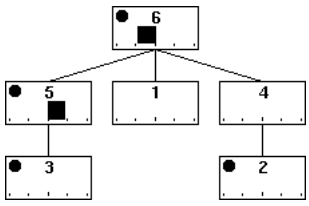


Figure 15.7. *The difference from figure 15.6 is that the Update Button has been pressed. The conflicts have been calculated using the new propagation algorithm, and the conflicts are shown as bars in the clique nodes below the label of the clique.*

If you have a careful look at the clique boxes you will notice 5 small ticks below the label of the clique, and that the name of the clique is not positioned in the middle of the box. The space here is used for the *conflict bars*. In figure 15.6 findings had been entered. In figure 15.7 Update has been pressed. The

conflicts have been calculated using the new propagation algorithm and the bars in the cliques indicate the conflicts.

The conflict measure of a clique is a number *n*, where $n > 0$. To show the monitor bars we calculate a new value $m = log2\ n$ (this is equal to *conf*, declared in section 15.5.1). The *m* value is what is shown using the bars. 0 is located at the center of the node, negative values to the left and positive to the right (in case of a color display left bars are red and right bars green).

Using the Weight Choice (see figure 15.8) the maximal left and right value of the ticks can be controlled. Setting Weight to e.g. 2 the ticks will indicate -2, -1, 0, 1, and 2.

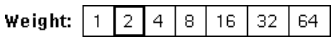


Figure 15.8. *The Weight Choice. Choose the shown range in the conflict bars.*

In the Pane Menu (see figure 15.2) the second item is Conflicts. Activating this will print the values of the conflicts in the Message Log Window (see figure 15.9). The Message Log Window will automatically be opened if necessary.

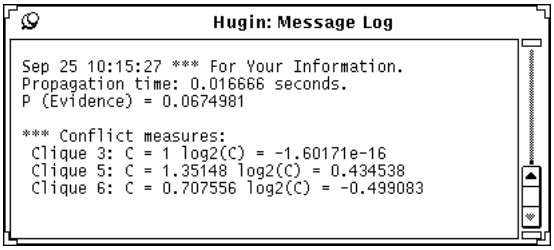

Hugin: Message Log
Sep 25 10:15:27 *** For Your Information.
Propagation time: 0.016666 seconds.
P (Evidence) = 0.0674981
*** Conflict measures:
Clique 3: C = 1 log2(C) = -1.60171e-16
Clique 5: C = 1.35148 log2(C) = 0.434538
Clique 6: C = 0.707556 log2(C) = -0.499083

Figure 15.9. *The conflicts message. Activating the Conflicts Item from the Pane Menu will print the conflicts into the Message Log Window.*

16

The Case Browser

The Case Browser (see figure 16.1) allows you to save a set of findings (a *case*) to be reused later. This is useful if you often want to start from the same set of findings, but do not want to save the entire domain (see the description of the Save Domain item in the File Menu, section 9.3). Saving just the findings are faster and takes less disk space than saving the complete domain, but requires a propagation after reloading the case. It is also possible to manipulate the case files from outside of Hugin Runtime, see section C.2 for details.

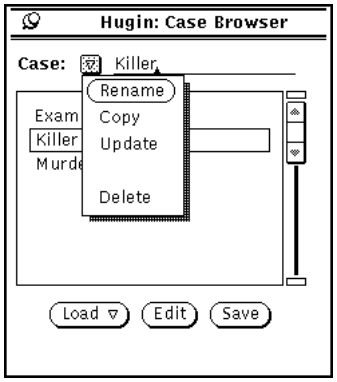


Figure 16.1. *The Case Browser*

The list of cases displayed on the scrolling list is associated with a directory, not a specific CPN. That is, when you change directory using the Directory Item in the CPN Browser (see section 10.1), the Case Browser will be updated to display the cases associated with the new directory. By associating the cases with the directory rather than the CPN, you are allowed to use the same cases for different CPNs. This is useful when CPNs are variations of each other, and thus shares some nodes and state names.

16.1 The Case Item

The Case Item consist of an abbreviated menu button (whose menu is displayed on figure 16.1), a text field, and a list of cases. You can select a case from the Case List by clicking SELECT on an item in the

list. This operation will copy the name of the selected case to the Case Text Field. You can now rename the case by editing the Case Text Field, and pressing RETURN. If you enter a name in the Case Text Field without selecting a case from the list, a case containing the current findings in the Base Window will be created.

The Case Menu contains a number of functions to manipulate the entries in the Case List. Except for Update, they all operate on the selected case.

Rename Rename the selected case to the name in the Case Text Field.

Copy Copies the selected case. The copy will have the name currently in the Case Text Field.

Update Reads the contents of the Case List from disk. Useful if other programs are adding or removing cases from the case directory.

Delete Delete the selected case.

16.2 The Case Buttons

The case buttons allows you to load, edit, and save cases. After loading, you can chose to propagate as illustrated by figure 16.2.

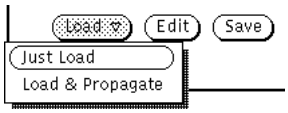


Figure 16.2. *The Case Load Menu*

Just Load Enter the findings specified in the selected case into the CPN in the Base Window.

Load & Propagate Enter the findings specified in the selected case into the CPN in the Base Window, and propagate with the new evidence.

Edit Edit the selected case textually. See section C.2 for a description of the case file format.

Save Save the findings entered into the Base Window into the case, which will have the named specified in the Case Text Field.

17

The Printer Window

The Printer Window allows several ways of printing and even previewing the CPN currently displayed in the Base Window. The Printer Window can only be used when the Base Window is in Test Mode. The window contains a number of controls determining the layout of the CPN. The most significant distinction if controlled by the Destination Item, specifying whether you want the output to go to a file or a printer. As illustrated by figure 17.1, the available controls depends on this choice.

17.1 Printing on a Printer

When the Destination Item has been set to Printer, the Printer Item specifies which printer to use (see figure 17.1–A). The menu associated will give a list of all printers available (that is printers specified in the `/etc/printcap` file). Selecting one of the printers in this menu will automatically copy this to the line. Only PostScript printers are currently supported.

The Format Item specifies the size of the paper used by the selected printer. Currently 5 standard sizes can be selected: A4, A3, A5, USLegal, and USLetter:

A4 21.022 * 29.73 centimeters, with a 1 centimeter margin in both directions.

A3 29.73 * 42.044 centimeters, with a 1 centimeter margin in both directions.

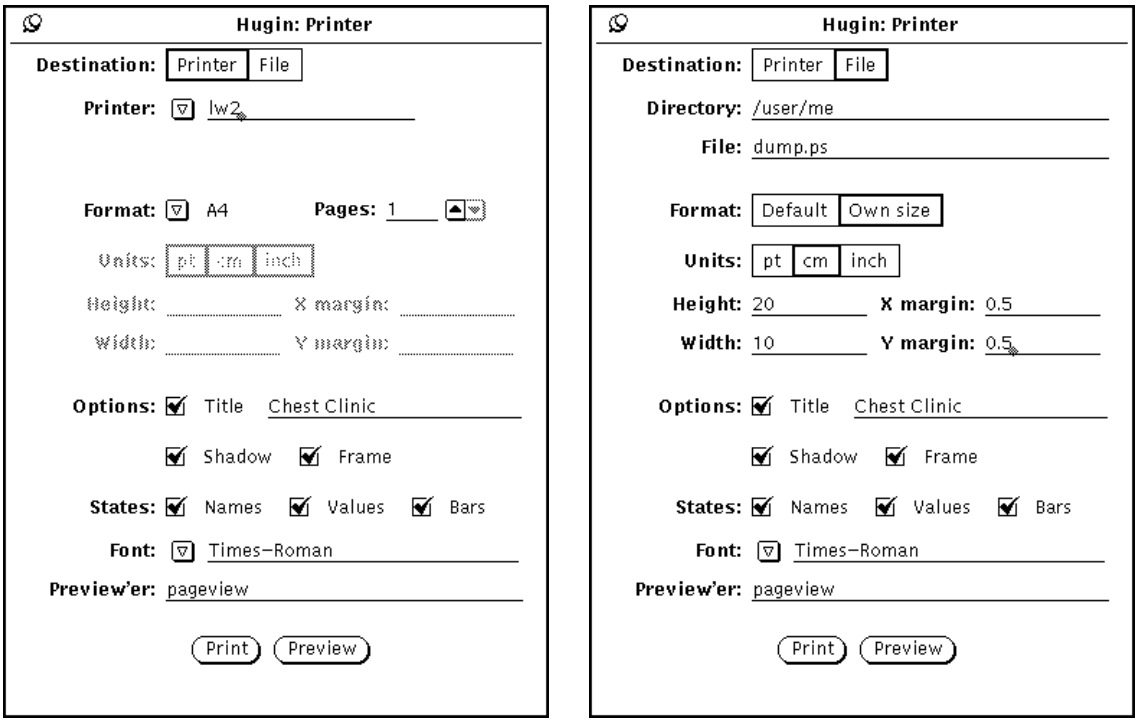
A5 14.865 * 21.022 centimeters, with a 1 centimeter margin in both directions.

USLetter 8.5 * 11.0 inches, with 0.25 and 0.04 inches borders.

USLegal 8.5 * 14.0 inches, with 0.89 and 0.5 inches borders.

The output will automatically be scaled to fit the selected format. The print will be rotated if this will produce a larger print. The last entry in this menu is Own Size for specifying an alternative size of the print (see later in section 17.4 for a discussion on this).

More than one page of output can be specified using the Pages Item. The output will try to fit the print into the number of pages specified, by rotating the paper and print and trying several ways of arranging



A B

Figure 17.1. *The Printer Window. In the first window, A, the output will go to the printer "lw2". In the second, B, the output will go to the file* `/user/me/dump.ps`*. In B the size of the output has been set using the Own Size option of Format.*

the pages next to each other. The total number of pages that was actually used will be shown in the lower right corner of the window.

When printing to more than one page ticks will be added in the corners of the print. These ticks are guidelines for cutting the pages before gluing the pages into one large poster[1].

When sending the print to a printer the print may be rotated to get a bigger print. Rotation will *not* happen when Own Size has been selected.

17.2 Printing to a File

When File is chosen in the Destination Item (see figure 17.1–B), the Directory and File text items are used to specify the file to store the resulting print in.

The format can now be specified in only two ways: Default or Own Size (see later in section 17.4 for a discussion on Own Size). If output is to be produced in Encapsulated PostScript (used with other programs to import figures, e.g. in LaTeX documents), Default should be used. The print shown in figure 17.2 was produced this way.

When printing to a file the print will *not* be rotated for better fitting.

[1] Scissors, glue and adhesive tape not provided.

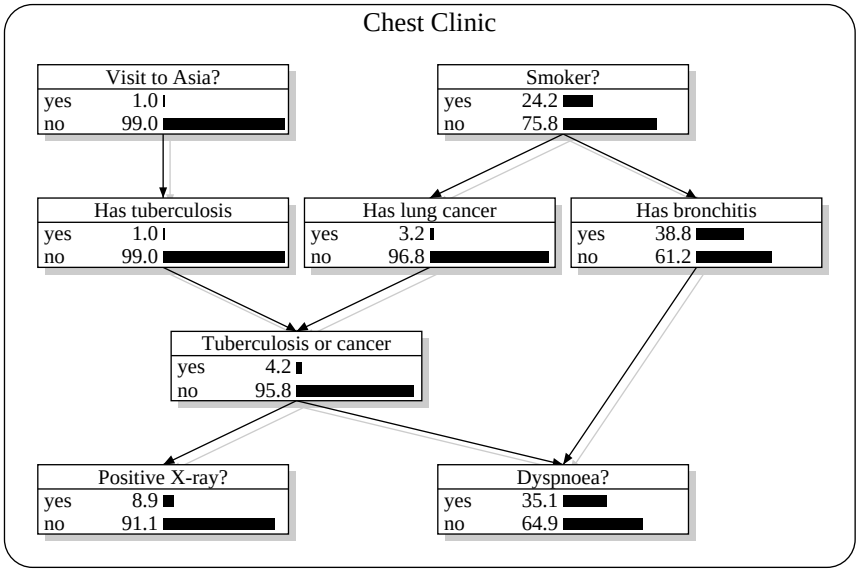


Figure 17.2. *Output from the Hugin printer. The Chest Clinic example. The options used are the same as in figure 17.1.*

17.3 Other Printer Options

There is a number of options controlling the final output.

Title If selected the title entered on the line next to the choice will be printed on top of the print.

Shadow A shadow of the arrows and boxes will be included. This will provide a more 3D look.

Frame Will draw a frame round the print.

All of these three options were set when the print in figure 17.2 was generated.

The States Options control how the states are draw. If *none* of the options are selected the nodes will be printed as a box with the label positioned in the middle of the box. If one ore more of the States Options are selected the boxes will be printed in a slightly different way.

First the maximal number of states in the nodes, *m*, is found. The label of the node will be positioned in a smaller box, where the height is *1/(m+1)* of the total height of the node box. The states will be centered in the left over area of the node box each having *1/(m+1)* of the total height of the box to be printed in.

Now we have a problem. If *m* is large *1/(m+1)* will be small, or to put it another way: if one of the nodes have many states it will be hard to read anything in the node boxes as all these states have to be printed. Note that the height and width of the boxes are the same as the height and width set for the domain when it was edited. One solution to the problem is to make the boxes higher and recompile the CPN.

17.4 Own Size of Output

When selecting the Own Size, the following items will be activated: Units, Height, Width, X margin, and Y margin.

The Units Item controls if the numbers entered should be read as points (1 point = 1/72 inch), centimeters, or inches.

The Height and Width Items controls the height and width of the output.

A laser writer is not able to print all the way to the edge of the paper. Using the X margin Item and the Y margin Item the margins of the paper (from the edge to where the printing can start) are set.

When Own Size has been selected with the destination set to Printer the print will *not* be rotated to fit better.

17.5 Previewing

The print can be previewed using the Preview Button. This will start the program specified on the Preview'er text item. If the Destination Item is set to Printer a temporary file is created to hold the PostScript and this file is later removed. If it is set to File the file will be created and then previewed (this means that an additional push on the Print Button will just produce a file similar to the one already created).

The default program to run is `pageview`, note however that this program will only run under OpenWindows, not under the MIT sample implementation of X11.

18

The Likelihood Window

The Likelihood Window (see figure 18.1) allows you to enter relative likelihoods for the individual states of node. This is different from entering findings, since ordinary findings only allows you to indicate whether a state is possible or not.

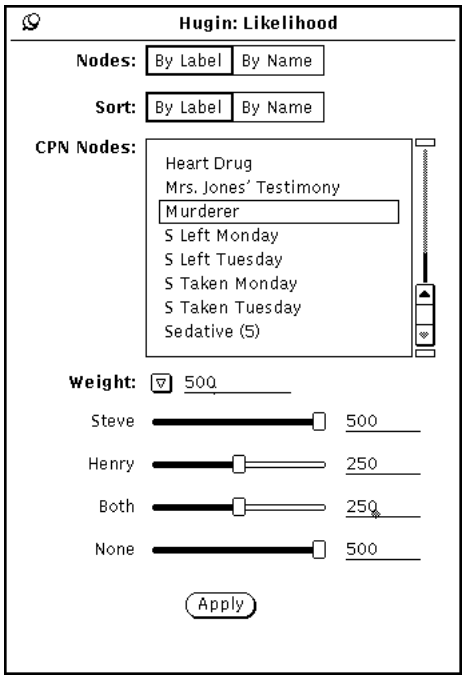


Figure 18.1. *The Likelihood Window*

The Likelihood Window is only active when the Base Window is in Test Mode.

18.1 The Nodes Items

The items at the top of the window allows you to choose a node. The Nodes Choice and Sort Choice control the display of the CPN Nodes Scrolling List, where you select a node.

Nodes Choose whether you want to display the names or the labels of the nodes in the CPN Nodes Scrolling List.

Sort Choose whether you want to sort the CPN Nodes Scrolling List by name or by label.

CPN Nodes Choose a node from the list by clicking SELECT on it.

When you have chosen a node, you enter the relative likelihood of the states in the node.

18.2 The Weight Item

You can enter what the maximum likelihood is considered to be, either by editing the Weight Text Field directly, or choosing one of the predefined values from the Weight Abbreviated Menu.

The weight has no influence on Hugins computations, only the relative size of each state matters.

18.3 The Likelihood Values

You can modify the relative likelihoods either by dragging the sliders, or by changing the numbers directly. It is possible to enter likelihood values greater than the weight, but it is unlikely to produce useful results. Setting the likelihood for one or more states to 0 implies a (possibly multistate) finding for the remaining states.

18.4 The Apply Button

Activate the Apply Button to notify Hugin about the current relative likelihoods of the states. Activating the button multiple times do have an effect, namely making the most likely states more and more likely.

19

The Tool Properties

You can customize Hugin Runtime from the Tool Properties Window available by activating the Tool Item from the Properties Menu.

Since there are more properties than can be shown on the screen at once, the Tool Properties have been divided into a number of categories. To select a category press MENU on the Category Abbreviated Menu Button (see figure 19.1), drag the pointer to the category you want to edit, then release MENU.

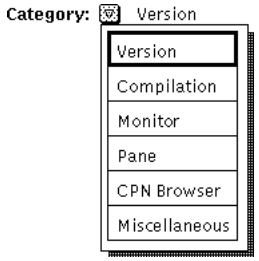


Figure 19.1. *The Category Menu*

When you have changed the properties in one or more of the categories, you must activate the Apply Button to make the changes take effect. Normally, the changes will only be effective until you leave the application. To make the changes permanent, i.e. also present next time you start Hugin Runtime, activate the Set Defaults Button.

If you want to undo any changes you have made to any category in the Tool Properties Window, you can use the Reset Menu Button illustrated in figure 19.2.

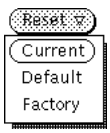


Figure 19.2. *The Reset Menu*

Current Reset the settings in the Tool Properties Window to the currently active settings. That is, undo all changes made since last time you activated Apply.

Default Reset the settings in the Tool Properties Window to the default settings Hugin Runtime has on start up. That is, undo all changes made since last time you activated Set Default. You will need to activate Apply to make the default settings active.

Factory Reset the settings in the Tool Properties Window to the factory settings. That is, undo all changes made since Hugin Runtime was installed. You will need to active Set Default to make the factory settings default next time you start Hugin Runtime. You will also need to activate Apply to make the factory settings active.

Please not that the Apply, Set Defaults, and Reset Buttons affects all categories. It is *not* possible to apply or reset a single category.

19.1 The Version Category

The Version Category, illustrated by figure 19.3, displays information about the version of Hugin Runtime. You can not change any properties in this category.

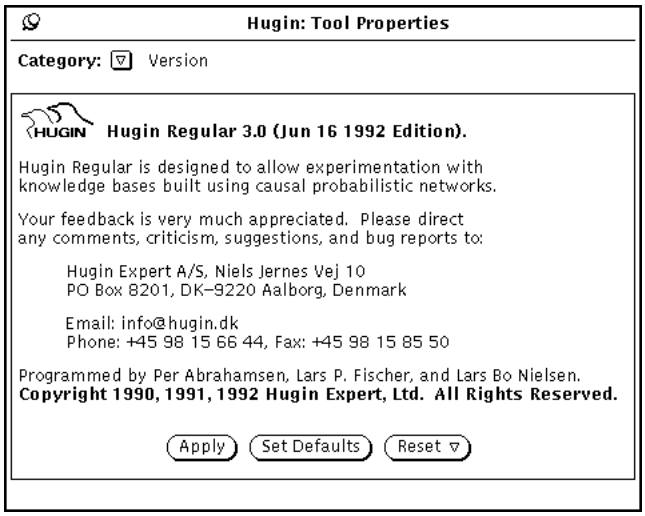

Hugin: Tool Properties
Category: Version
HUGIN
Hugin Regular 3.0 (Jun 16 1992 Edition).
Hugin Regular is designed to allow experimentation with knowledge bases built using causal probabilistic networks.
Your feedback is very much appreciated. Please direct any comments, criticism, suggestions, and bug reports to:
Hugin Expert A/S, Niels Jernes Vej 10
PO Box 8201, DK–9220 Aalborg, Denmark
Email: info@hugin.dk
Phone: +45 98 15 66 44, Fax: +45 98 15 85 50
Programmed by Per Abrahamsen, Lars P. Fischer, and Lars Bo Nielsen.
Copyright 1990, 1991, 1992 Hugin Expert, Ltd. All Rights Reserved.
Apply
Set Defaults
Reset

Figure 19.3. *The Version Category*

19.2 The Compilation Category

The settings in the Compilation Category, illustrated by figure 19.4, allow you to control the compilation of a CPN. You can choose between different strategies affecting both the compile time and the run time, depending on the structure of the CPN.

19.2.1 Triangulation

You can choose between three different triangulation algorithms. A good triangulation is the key to an efficient compiled domain. Unfortunately, triangulation is a very time consuming process (it is NP complete). It is therefore necessary to use some heuristics in the process. Which heuristics work for a given CPN is difficult to tell, so some experimentation is advisable.

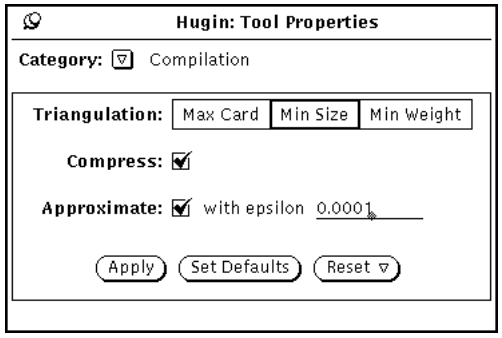


Figure 19.4. *The Compilation Category*

Max Card A very simple algorithm that is well suited for small CPNs. Do not try to apply this algorithm on large CPNs, unless you for some reason require deterministic results.

Min Size The algorithm which generally gives the best results. The algorithm uses a random factor in the triangulation, so you may try to compile several times, in order to get the best result.

Min Weight An algorithm which sometimes gives good results on some large CPNs. This algorithm also uses a random factor in the triangulation.

Note that **min-size** and **min-weight** are heuristic methods using a level of randomness in execution. They might therefore not give the same result in two successive runs. If obtaining an optimal result is important, several compilations using one of the heuristic methods can be done, keeping the best one obtained (eg. the one resulting in the smallest domain).

The **max-card** method is currently the only method that is non-heuristic, ie. it will the same triangulation on every run. Regrettably, the results obtained using maximum cardinality search are not very good. It should be used only when repeatability is important.

See [Kjæ90] for a detailed discussion of the various triangulation algorithms,

19.2.2 Compress

After compilation, and possible approximation, the domain will be compressed. Compressing can drastically reduce the memory usage, as well as improving the speed of propagations.

Most of the memory used for a compiled domain is used to store the belief tables for the cliques in the junction tree. Many of the entries in the belief tables will be zero, reflecting the fact that these combinations of occurrences in the model are highly unlikely, e.g. will never happen in the real world. To conserve memory, these data elements can be removed, making the tables smaller. This is called compression.

Note also, that compression is only possible after compilation is completed. This means that enough memory to store the full domain must be available while compiling.

It is also important to note that no data is lost from the knowledge base and that the reasoning done in Hugin does not become less precise because of compression. The zero data elements will not contribute to the propagation process and removing them will therefore not affect the results obtained.

The amount of compression will be reported after the compilation has succeeded. This value is given as the size of the compressed tables in percent of the size of the original tables. When a domain is compressed, the overhead of operation becomes larger, as does the storage overhead. This is because the

tables are changed into sparse matrices, making it necessary to keep indices these matrices. As a rule of thumb, compression is only an advantage if the compression factor is better than 50%.

19.2.3 Approximate

To improve the effect of compression, you can choose to treat small probabilities as zero. By doing this, you will remove the “rare cases” from the CPN. This might or might not be acceptable, depending on the application. The Epsilon Field affects how much probability mass can be removed, i.e. how rare a case should be before it is removed. See [JA90] for a detailed description of both the algorithm and the theory behind it.

Note that it is very hard to give a theoretical description of the effects of approximation on the reasoning process. In most cases, approximation can be done without affecting the belief values as long Epsilon is on the order of 0.0001. For many domains even coarser approximations can be done without severely affecting the results obtained. However, it cannot be guaranteed that approximation with even very small values of Epsilon will not give wrong results.

The probability mass after the approximation will be reported after the compilation has succeeded.

When using approximation, compression should also be used.

19.3 The Monitor Category

The monitor properties (see figure 19.5) affect the look of the monitor windows described in chapter 11. You can set the size of the components of a monitor, as well as the font and precision used. If you are using a color screen, additional items (not shown on the figure) will appear, allowing you to set the colors of the bars in the monitor windows.

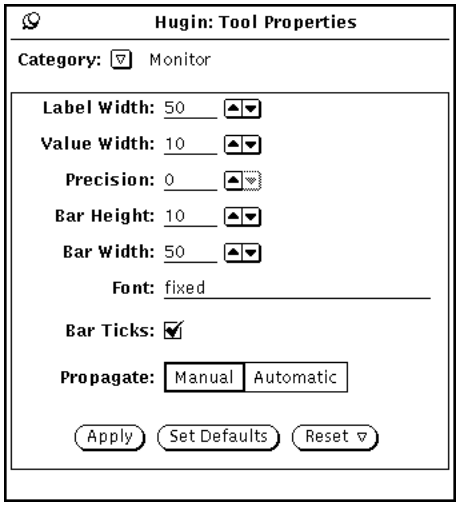


Figure 19.5. *The Monitor Category*

Label Width The horizontal space reserved for the state labels in the monitor windows, measured in screen pixels.

Value Width The horizontal space reserved for the numeric state value in the monitor windows, measured in screen pixels.

Precision The precision of the numeric state value in the monitor windows. Zero indicates no digits to the left of the decimal point.

Bar Height The height of the bars in the monitor windows, measured in screen pixels.

Bar Width The width of the bars in the monitor windows, measured in screen pixels.

Font The name of the font used in the monitor windows. You can get a list of available fonts by typing the command '`xlsfonts`' in a shell window.

Bar Ticks Check to insert tick marks for the bars in the monitor windows.

Propagate Choose 'Automatic' to make propagation occur automatically when new evidence is entered through the monitor windows.

19.4 The Pane Category

The pane properties (see figure 19.6) affect the look and feel of the pane in the Base Window, more precisely, how dragging nodes feels in Modify Mode, and what font to use for the node labels.

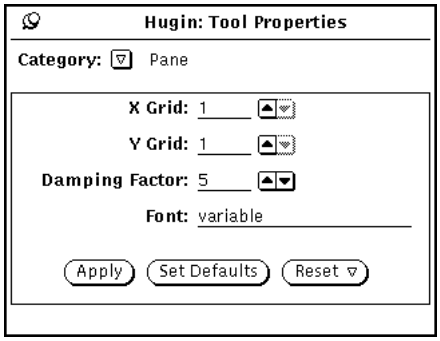


Figure 19.6. *The Pane Category*

19.4.1 Grid

To make it easier to align nodes in the CPN, you can set a grid. When you drag a node in Modify Mode, the node will be aligned to the X and Y coordinates specified with the grid properties.

Warning: The grid is currently specified in screen pixels, making the grid very fragile. An improved version based on CPN points is being worked on.

You can specify both the X and Y grid.

19.4.2 Damping Factor

When you press SELECT in Modify Mode, it might be because you want to select a node, or because you want to drag the node to another location. In order to avoid accidently moving a node when the intention was to select it, a drag will not be initiated before you have moved the pointer some distance. The Damping Factor determines the distance measured in screen pixels the pointer must be moved before a drag is initiated.

19.4.3 Font

The name of the font used for node labels in the pane of the Base Windows. You can get a list of available fonts by typing '`xlsfonts`' in a shell window.

19.5 The CPN Browser Category

The CPN Browser Category contains a message directing you to use the CPN Browser Window for customizing the Directory Menu, as illustrated on figure 19.7.

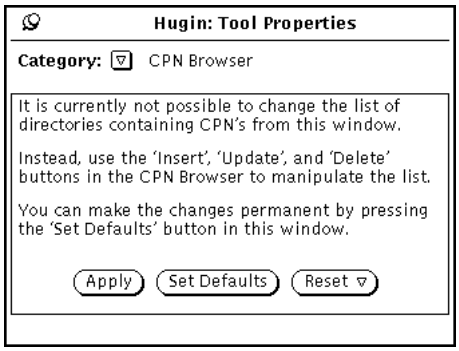

Hugin: Tool Properties
Category: CPN Browser
It is currently not possible to change the list of directories containing CPN's from this window.
Instead, use the 'Insert', 'Update', and 'Delete' buttons in the CPN Browser to manipulate the list.
You can make the changes permanent by pressing the 'Set Defaults' button in this window.
Apply
Set Defaults
Reset

Figure 19.7. *The CPN Browser Category*

19.6 The Miscellaneous Category

The Miscellaneous Category (see figure 19.8) contains properties which does not fit into any of the other categories.

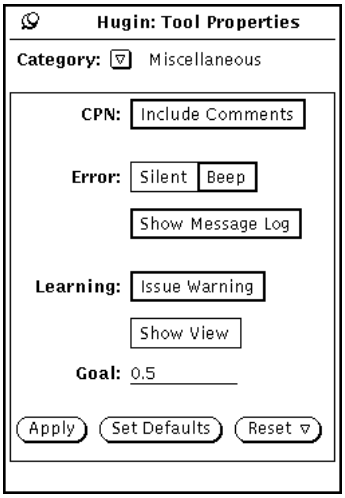


Figure 19.8. *The Miscellaneous Category*

19.6.1 CPN

Specify the format of the CPNs.

Include Comments Choose this item to include comments in the CPN files created by Hugin Runtime when you save the CPN in Build or Modify Mode.

19.6.2 Error

Specify what happens when an error occurs.

Beep Make a sound.

Silent Stay silent.

Show Message Log Show and raise the Message Log Window.

The error message will always be shown in the left footer and appended to the Message Log Window. See section 20.2.

19.6.3 Learning

What happens when you start a batch learning job (by activating the Learn Button, as described in section 14.6).

Issue Warning Ask if you really want to overwrite the old probability tables, now that you are learning new ones.

Show View Show the Learning View Window, described in section 14.7.

Goal Draw a line in the Learning View Window indicating what you expect the result to be.

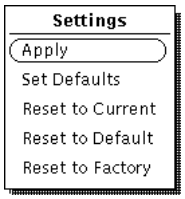


Figure 19.9. *The Settings Menu*

19.7 The Settings Menu

It is also possible to Apply, Set Default, or Reset the Tool Properties from the Settings Menu (figure 19.9), which is available by pressing MENU in the background area of the Tool Properties Window. The Settings Menu is available in all the categories.

20

Help, Errors, and Other Information

Hugin Runtime has a number of windows and window areas designated for providing various kind of information, most significantly context sensitive help and error messages.

20.1 Help

All the facilities in Hugin Runtime are described through the OPEN LOOK context sensitive help facilities. To get help on any part of Hugin Runtime, place the mouse pointer above the item, and press the ‘HELP’ key. A window containing a magnifying class and a help text will appear, as illustrated by figure 20.1.

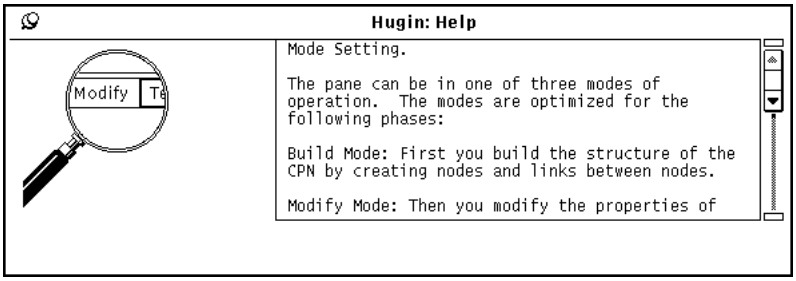

Hugin: Help
Modify
Mode Setting.
The pane can be in one of three modes of operation. The modes are optimized for the following phases:
Build Mode: First you build the structure of the CPN by creating nodes and links between nodes.
Modify Mode: Then you modify the properties of

Figure 20.1. *A Help Window*

On keyboards without a ‘HELP’ key you must use another key, typically ‘F1’. The context sensitive help requires that the ‘Hugin.info’ file and the ‘HUGINHOME’ environment variable are set up properly.

20.2 Error Messages

Error messages will be shown in the left footer of the appropriate window, as illustrated by figure 20.2. The appropriate window is the window where you activated the operation which caused an error. The error message will disappear as soon as you click with the mouse or type a key in any window controlled by Hugin Runtime.

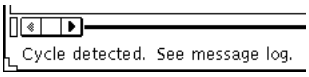


Figure 20.2. *The Left Footer*

Error messages will also be displayed in the Message Log Window, accessible from the View Menu in the Base Window (see section 9.4). The Message Log Window, as illustrated by figure 20.3, contains a log of all error messages and warnings, often including a more detailed explanation than would fit the left footer. Other more or less useful information is occasionally also shown in the Message Log Window.

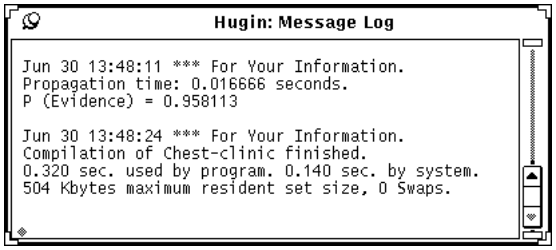

Hugin: Message Log
Jun 30 13:48:11 *** For Your Information.
Propagation time: 0.016666 seconds.
P (Evidence) = 0.958113
Jun 30 13:48:24 *** For Your Information.
Compilation of Chest-clinic finished.
0.320 sec. used by program. 0.140 sec. by system.
504 Kbytes maximum resident set size, 0 Swaps.

Figure 20.3. *The Message Log Window*

The Miscellaneous Category in the Tool Properties Window (see section 19.6) allows you to specify that Hugin Runtime should beep in case of an error, and if the Message Log Window automatically should be shown.

20.3 CPN Status

The right footer of the Base Window is used for status information, as suggested by OPEN LOOK. In Hugin Runtime the usage of the right footer is different in Test Mode than in Build Mode or Modify Mode.

20.3.1 Build Mode and Modify Mode

If the CPN contains a cycle the message “Cyclic” is displayed in the right footer, as illustrated by figure 20.4.

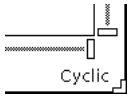


Figure 20.4. *The Right Footer*

Similarly, if the CPN contains unconnected subnets the message “Unconnected” is displayed in the right footer. Hugin Runtime refuses to switch to Run Mode if the CPN is either unconnected or cyclic.

However, you are still allowed to save the CPN, just be careful not to try to compile the CPN from the CPN Browser, since the compiler does not try to validate the CPN.

20.3.2 Test Mode

In Test Mode, the right footer displays general information about the evidence entered into the CPN. The possible messages includes:

Initialized The CPN has just been initialized, thus no evidence has been entered.

P (Evidence) = *all* All evidence is new, or some of the old evidence has been retracted. The probability of the evidence is *all*.

P (New) = *new* **— P (All) =** *all* New evidence has been entered, and no evidence has been retracted. The conditional probability of the new evidence given the old evidence is *new*. The joint probability of the new and old evidence is *all*.

Likelihood Part of the evidence is from entered likelihood. This makes it impossible to calculate the probability of the evidence.

Adapted The CPN has just been adapted to the evidence entered. This implies an initialization.

20.4 Compilation Log

The Compilation Log Window, illustrated by figure 20.5, contains information about the internal structures created during compilation.

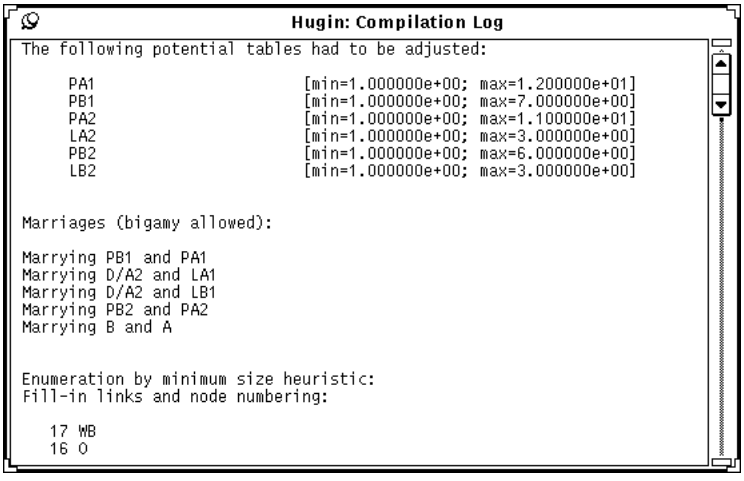


Figure 20.5. *The Compilation Log*

The information in compilation log includes the number of cliques generated, to which cliques a node is assigned, the size of the tables, and the configuration of junction tree. Advanced users can evaluate the efficiency of the compiled CPN with this information.

20.5 The Demonstration Directory

The Directory Menu (see section 10.1) in the CPN Browser has a special entry named:

```
<Demonstration>
```

The Demonstration directory contains CPNs built into Hugin Runtime. The built-in CPNs have no editable or runable form, you will need to compile them each time you want to try them out. However, they have an extra property associated with them, namely a story. You can use the Story Item in the View Menu (see section 9.4) to get a window containing their story, as illustrated by figure 20.6.

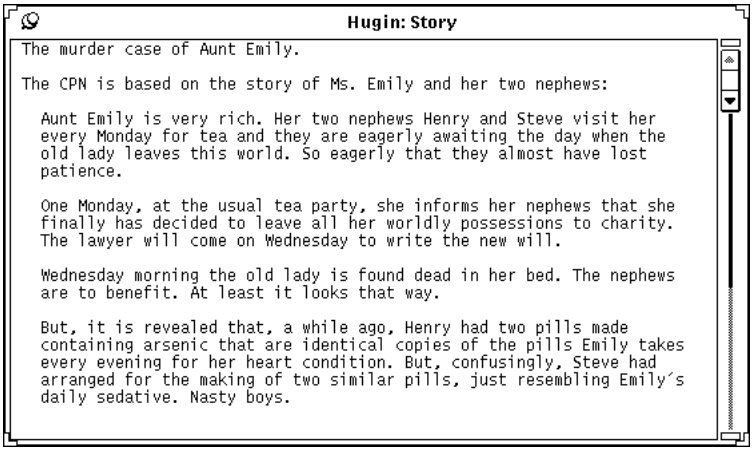

Hugin: Story
The murder case of Aunt Emily.
The CPN is based on the story of Ms. Emily and her two nephews:
Aunt Emily is very rich. Her two nephews Henry and Steve visit her every Monday for tea and they are eagerly awaiting the day when the old lady leaves this world. So eagerly that they almost have lost patience.
One Monday, at the usual tea party, she informs her nephews that she finally has decided to leave all her worldly possessions to charity. The lawyer will come on Wednesday to write the new will.
Wednesday morning the old lady is found dead in her bed. The nephews are to benefit. At least it looks that way.
But, it is revealed that, a while ago, Henry had two pills made containing arsenic that are identical copies of the pills Emily takes every evening for her heart condition. But, confusingly, Steve had arranged for the making of two similar pills, just resembling Emily´s daily sedative. Nasty boys.

Figure 20.6. *The Story Window*

Apart from the built-in CPNs, you can create your own CPNs in the Demonstration directory. These CPNs can be edited and compiled as usual, but they will not be stored in the file system. You can gain a little speed by not going through the file system, at the cost of loosing the CPNs when you leave Hugin Runtime.

The Demonstration directory is primarily intended to be used in Hugin Demonstration, where all file operations are disabled. To learn more about the difference between Hugin Demonstration and Hugin Regular, read the preface to this manual.

Part III

Appendices

A

The Built-in Examples

This appendix provides descriptions of the four example CPNs distributed with Hugin Runtime. For each example you will find a brief description of the problem area modeled by the CPN. This is followed by an explanation of the structure of the CPN, i.e. why the specific model of the problem area was chosen. Finally, a few experiments that can be made with the example are suggested.

You are encouraged to play with these examples to get a feel for how Hugin Runtime operates and how domain models are created. Even though the examples are fairly small, they illustrate the basic concepts of causal probabilistic networks and give examples of some of the strengths of the method.

A.1 The Chest Clinic Example

The Chest Clinic example is also described in the tutorial part. The aim of the CPN is to diagnose patients at a Chest Clinic. The CPN discriminates between three diseases, bronchitis, lung cancer, and tuberculosis:

> Shortness-of-breath (dyspnoea) may be due to tuberculosis, lung cancer or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The result of a single chest X-ray does not discriminate between lung cancer and tuberculosis, as neither does the presence or absence of dyspnoea.

As can be seen from the description, we have two symptoms, a positive X-ray test and shortness of breath, and two risk factors for the diseases, smoking and a visit to Asia. We create a node for each disease, symptom, and cause. A disease node is made a child of the nodes for its risk factors and a parent of the nodes for its symptoms.

A complication arises because the X-ray test cannot discriminate between lung cancer and tuberculosis. We therefore introduce a node that corresponds to either of these diseases. This new node models the *appearance* of lung cancer and tuberculosis in a clinical test. The new node is made a child of the nodes for the diseases that creates the effect and a parent of the symptoms of either disease.

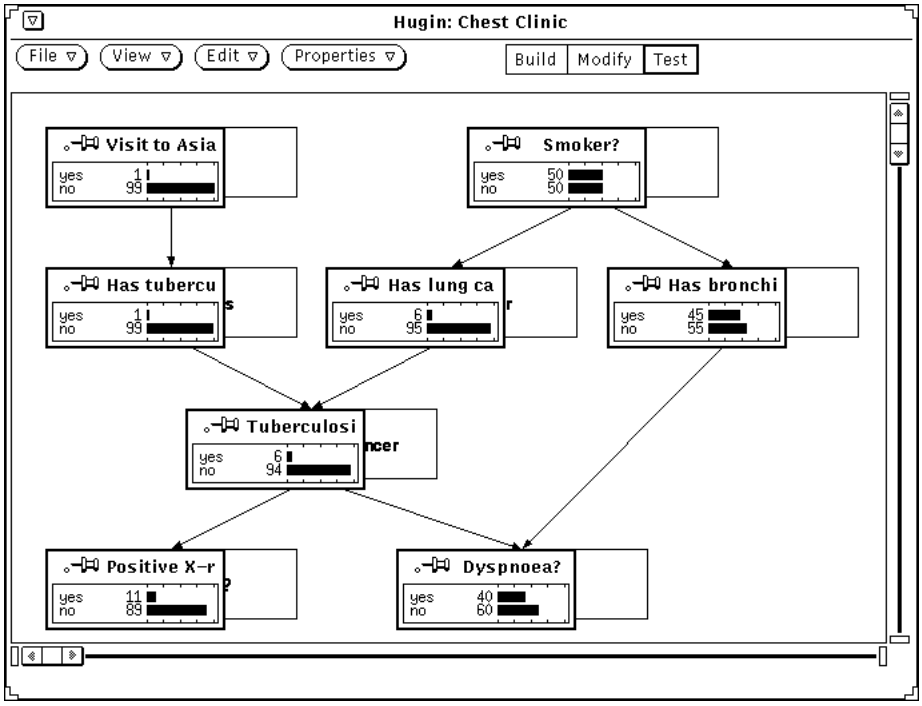


Figure A.1. *The CPN for the Chest Clinic example, including monitor windows displaying initial beliefs for the important nodes.*

Figure A.1 illustrates the CPN for the Chest Clinic example, with monitor windows displaying the initial beliefs. Usually, findings would be entered for the symptoms and risk factors and propagation would show the diagnosis as beliefs in the monitors for the disease nodes.

The addition of the extra node for tuberculosis or lung cancer is a common trick that can often be employed to more closely model the relationship of an effect (here the disease) to its effect (here the symptom). The trick can reduce complexity, computationally as well as conceptually.

A.2 The Reproduction Example

The Reproduction example is taken from genetics. The example concerns breeding and would typically be found within animal husbandry, but in order to make it more attentive, we will state it in terms of human beings:

> Florence and Gregory are about to reproduce. However, Gregory is Florence's nephew, and in the annals of Bartholomew, the father of Florence and grandfather of Gregory, a life-threatening disease has haunted. The disease is caused by a dominant allele A, and appears in a rather late stage of the individual's life-time. Bartholomew married twice, hence Florence and Gregory's mother are half–siblings. Neither Florence nor Gregory, their parents, nor Gregory's grandmother have shown any signs of the disease. What is the risk that their child will inherit the fatal characteristic.

To formulate the problem as a causal probabilistic network, we first draw the genealogical structure of the domain (see figure A.2).

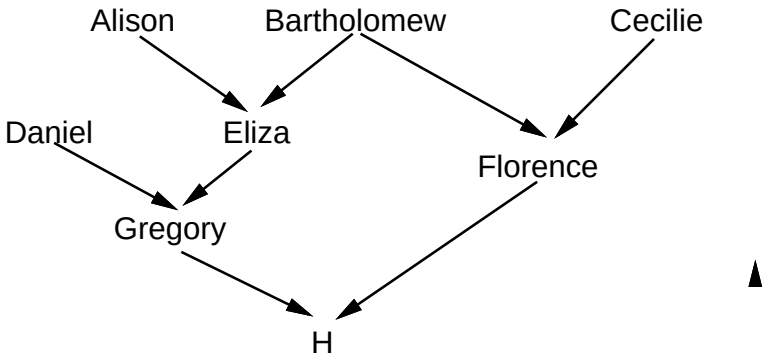


Figure A.2. *Genealogical structure for the reproduction example.*

Each individual is represented by name, and parent-child relations are represented by an arrow from parent to child. Note that the diagram contains two different paths from Bartholomew to H: via Eliza and Gregory, and via Florence.

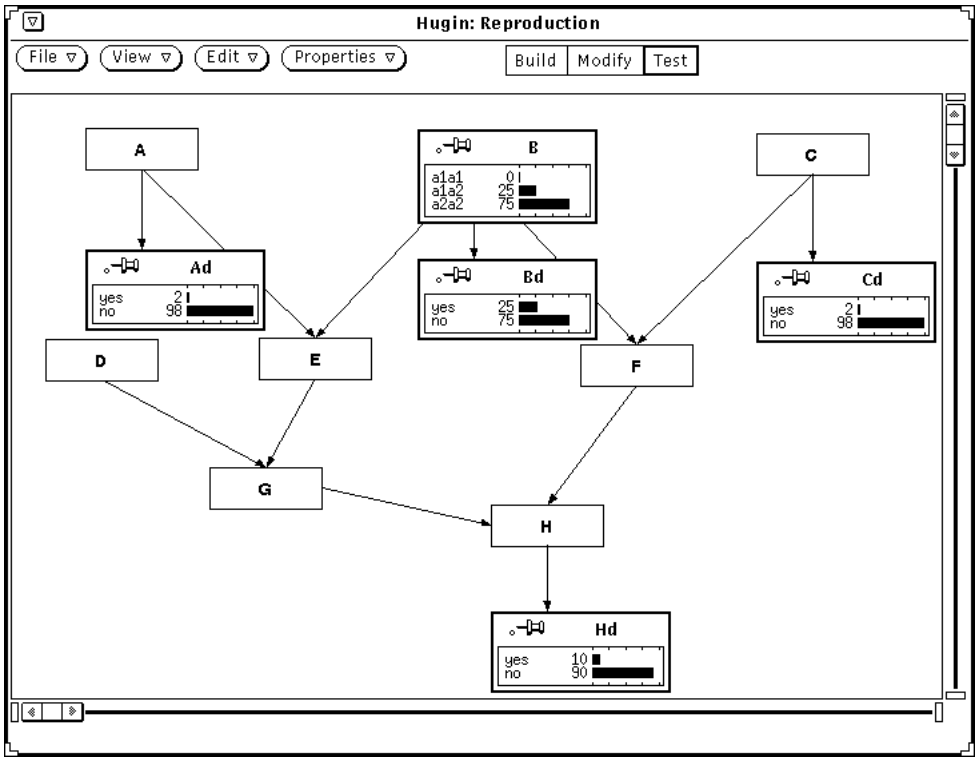


Figure A.3. *The CPN for the Reproduction example, including monitor windows displaying initial beliefs for the disease nodes.*

Now we must construct a CPN that models the inheritance of the fatal disease through the genealogical structure. In general, we are interested in determining the genotype of the individuals. To each individual, we associate a node labeled with his/her initial and having as states the possible genotypes identified by the combinations of alleles. Hence, each individual is of exactly one of the genotypes a1a1, a1a2, or a2a2.

In our case, where one allele is dominant, this means that the genotypes a1a1 and a1a2 are carriers of the disease, whereas a2a2 is not.

What can be observed is whether the disease is present or not, but this can only be determined when the individual has reached a mature age due to the sneaky character of the disease. In order to be able to enter relevant information on observed occurrences and ask for expectations of the disease, we add a disease node to the elder and the upcoming generation. These nodes, labeled with the individuals initial with index d, have two possible states, “yes” and “no”, corresponding to the presence or the absence of the disease, respectively.

With the disease nodes added, we obtain the CPN illustrated by figure A.3. In the figure, monitor nodes have been created for the disease nodes, as these nodes are where findings should be entered or, in the case of “Hd”, results obtained. For the three elders, it can be investigated if they have the disease. When a combination of disease findings for the elders have been entered and propagated the belief in the child developing the disease can be seen.

A single monitor window has been created for a genotype node (Bartholomew). This monitor window displays the states of these nodes. There is a state for each combination of the two genes a1 and a2, as described above.

By experimenting with this CPN, one can find the probability of the child developing the disease. The CPN could also be extended with disease nodes for the other members of the family. These nodes can be used to obtain the disease probabilities for these family members. Finally, the CPN could be extended with further family members as the family grows.

A.3 The Case of Aunt Emily

The Aunt Emily CPN is the largest and most complicated of the example CPNs supplied with Hugin Runtime. The CPN is based on the story of Ms. Emily and her two nephews:

> Aunt Emily is very rich. Her two nephews Henry and Steve visit her every Monday for tea and they are eagerly awaiting the day when the old lady leaves this world. So eagerly that they almost have lost patience.
>
> One Monday, at the usual tea party, she informs her nephews that she finally has decided to leave all her worldly possessions to charity. The lawyer will come on Wednesday to write the new will.
>
> Wednesday morning the old lady is found dead in her bed. The nephews are to benefit. At least it looks that way.
>
> But, it is revealed that, a while ago, Henry had two pills made containing arsenic that are identical copies of the pills Emily takes every evening for her heart condition. But, confusingly, Steve had arranged for the making of two similar pills, just resembling Emily’s daily sedative. Nasty boys.
>
> But, did they have the pills with them for tea? Did they use them? If they did, which of them killed her? Or did she just die of old age?
>
> Watch the evidence pile up. There is a possibility of performing an autopsy, of examining the remaining pills in the pill boxes and finding out the time of death.
>
> Also one can examine witnesses. E.g. did or didn’t the cleaning lady notice anything unusual with the medicine cupboard, that the door was not closed neatly? Did or didn’t Emily tell her friend Dora that “there were too many pills in her box with the sedative?”

The aim is now to build a CPN that will help determine whether Aunt Emily was murdered and if so, who did it. We want to establish the cause of the death of Aunt Emily, so we have a node for this. The possibilities are that she died from poisoning, from something else, or that she is still alive and well. Poisoning can be split into three: poison from the heart drug, from the sedative, or both. The cause of death node has five states, then.

There are two basic facts that we can obtain: was Aunt Emily alive on Tuesday and Wednesday, respectively? We have a node for each of these observations. The cause of death is a child of these nodes; whatever the nephews put in the pills there is no murder if Aunt Emily didn't die from it.

We then have to model the possible poisoning from the pills. For this, we create two similar structures, one for each kind of pill. Each structure has five nodes. First of all, we want to know if any poisoned pills where added to the regular pills. We create a node named after the kind of pill (e.g., "Heart Drug") representing the number of poisoned pills added to the regular pills of this kind (zero, one, or two), i.e. the node has three states.

There is then a possibility that Aunt Emily takes a poisoned pill from the set on Monday. This is a "yes" or "no" node, and the probabilities depend on the number of poisoned pills. Depending on the number of poisoned pills initially available and if Aunt Emily took one of them or not, zero, one, or two poisoned pills will remain after Emily has taken her pill on Monday. We add a node for this. The same pattern is repeated on Tuesday, with the dependence being on the number of pills remaining on Monday.

The result of the above is two sets of five nodes. To connect the nodes created so far, we establish the following: whether Aunt Emily is alive Tuesday depends on the possibility of her taking a poisoned pill on Monday. Her being alive Wednesday depends on the pills taken Tuesday, and of course whether she was alive at all on Tuesday. A poisoned pill can be taken Tuesday only if Aunt Emily is still alive, and the cause of death depends on the pills taken (both days) and Aunt Emily being alive (both days). The cause of death node ends up having six parents.

All we need to do now is to add nodes for the observations we can make and the conclusions we want to reach. To take the last item first, we want to know the identity of the murderer, if any. There are four outcomes: Steve did it, Henry did it, they both did it, or none of them did it. We create a node for this and make it a child of the cause of death node.

Then there are the results of investigation. We can do an autopsy to establish if there is any arsenic in the organs of aunt Emily. The outcome of the autopsy will depend on the cause of death. Finally, we can add nodes for the reports given by each of the witnesses. The report of the cleaning lady, Mrs. Jones, will depend on the initial number of poisoned pills; the more persons that have been at the medicine cupboard, the greater chance that she will say that she has seen something unusual. The report of Dora depends only on the initial number of poisoned pills among the regular sedative pills.

Note in the above that the observation nodes are children of other nodes, not parents. The observations we can make depends on the state of the model but they never influence it. Our observations will, on the other hand, influence our *beliefs* about the state of the model when no information is directly available. As an example, finding arsenic during autopsy will affect our belief in Steve or Henry being the murderer and in the number of poisoned pills initially available. It will even affect our beliefs (expectations) of the reports of Dora and Mrs. Jones if these have not yet been investigated.

In figure A.4, the CPN is displayed. The monitor windows for the observations we can make and the conclusion has been opened. Note that the number of poisoned pills left Tuesday can be observed by simply looking in the medicine and analyzing the pills left.

Try to enter observations into the network and see the conclusion reached. Note how the effects of findings ripple throughout the CPN, e.g. from the "Autopsy" node to the "Dora's Testimony" node.

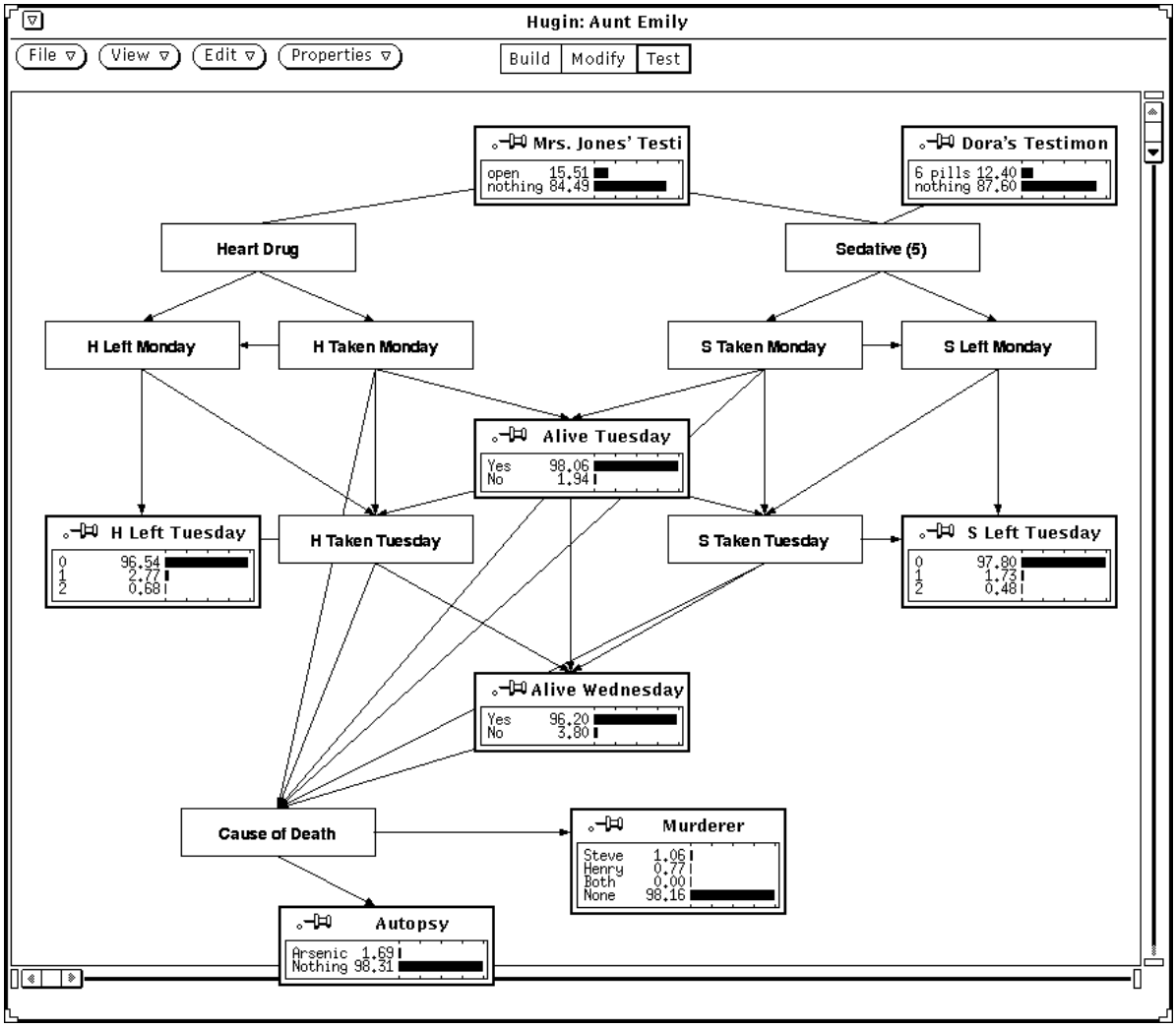


Figure A.4. *The AuntEmily CPN, including monitor windows displaying initial beliefs for the most important nodes.*

In figure A.5 a case is shown. Aunt Emily is dead, and as there are still two poisoned heart drug pills left, Steve is in trouble. So is Henry, of course, as he will be charged for attempted murder.

Experimenting with this network should be instructive as it displays many of the strengths and properties of the Hugin approach.

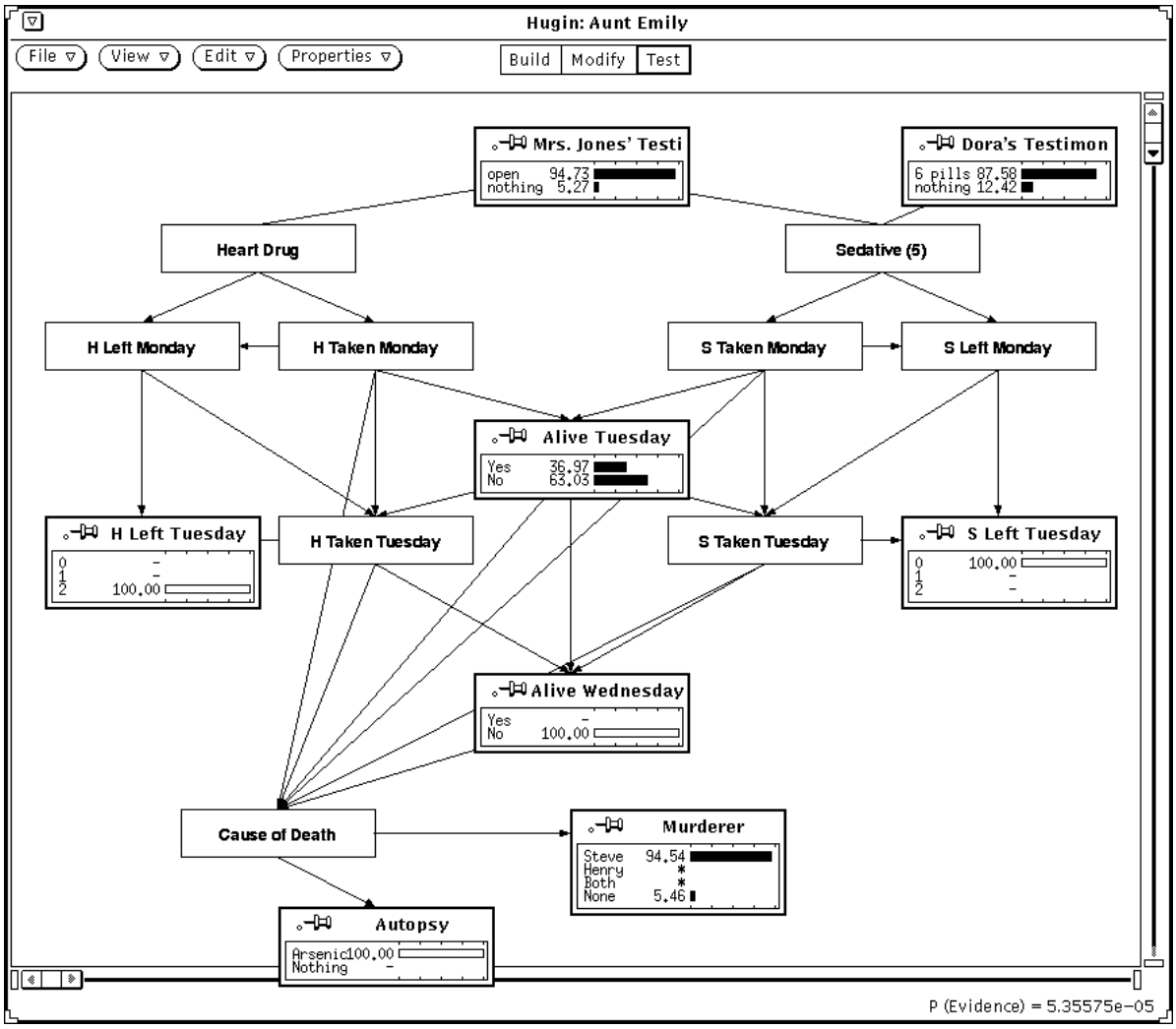


Figure A.5. *Aunt Emily has been murdered. Autopsy shows arsenic and it looks like Steve is the sinner.*

A.4 The Automobile Example

The automobile example performs basic fault diagnostics on a car. The components of the model are automobile parts such as a battery and a fan belt and measurable properties of the car such as engine temperature and oil pressure and important parameters such as “does the motor run” and “do the head lights work”.

Each component and property is represented by a node. Each node has a number of states, each state representing a particular state of the component or property. The “Engine” node has states “running” and “stopped”, the temperature node has states “normal”, “hot”, and “very hot”, and so on.

The links in the model show that a component or property is affected by some other component or property. For example, a broken fan belt will most likely cause the engine temperature to rise, while a broken electrical system will cause the head lights to fail. The complete network, including monitors displaying the initial beliefs of each node, is illustrated by figure A.6.

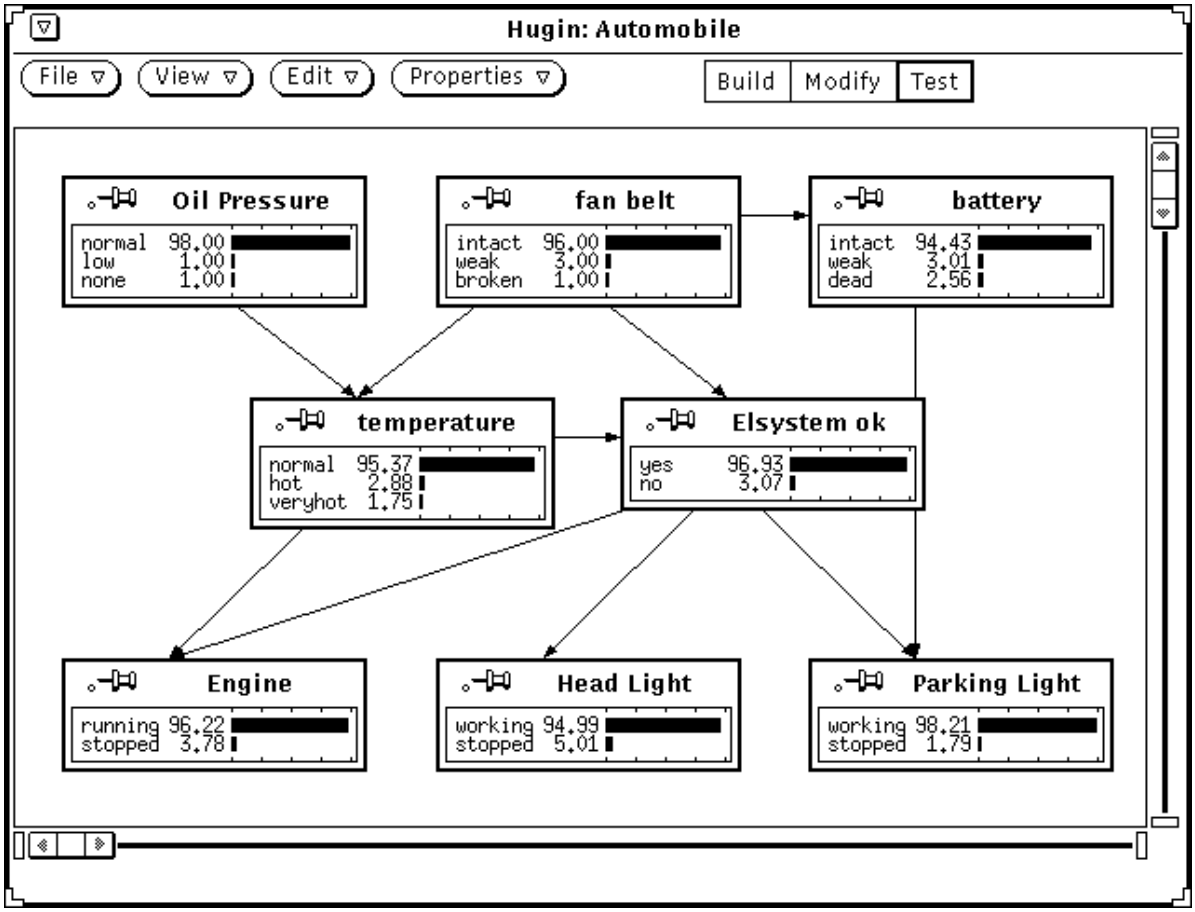


Figure A.6. *The Automobile CPN, including monitor windows displaying initial beliefs.*

Looking at the figure we see that the working of the engine depends on the temperature and the electrical system. The head light uses the electrical system, while the parking light will work from either the electrical system or the battery. The temperature depends on the oil pressure and on the fan, while the battery will only be charged if the fan belt is OK. Likewise the correct functioning electrical system depends on the fan belt, but it also depends on the temperature not being too high.

The Automobile CPN can be used in several ways. The most obvious is diagnostics; if the parking light of the car modeled fails, one can enter this observation as a finding and propagate. This leads to an

increase in the beliefs in the battery or the electrical system having failed. Having seen this, one could then investigate these two components and enter the results as new findings, and so on. This technique can help find the culprit of the problem.

The CPN can also be used for analysis; if you want to see the effect of a broken fan belt or a low oil pressure, these can be entered as findings and the effects seen. Such analysis can be used during design or maintenance to find out how critical a given component is, etc.

One might argue that the present model is a bit simplistic. To be useful for diagnostics or analysis of real automobile, a few more nodes and causal links will be needed. This is, as they say, left as an exercise for the reader.

B

The CPN Specification Language

In this manual, we have described how to create and edit a CPN using Hugin Runtime. However, in some cases you might prefer to edit the CPN specification by other means.

- You may want to use your favorite text editor allowing you to cut, paste, and copy nodes and tables in the CPN.
- For large networks, you probably want to use an independent program to automatically generate (parts of) the CPN.
- You may want to use the CPN from other programs than Hugin Runtime.

In order to allow this to happen, we have specified a textual format for the CPNs. The textual format is also supported by some other organizations.

When you edit a CPN from Hugin Runtime, you actually manipulate a CPN specification file. This CPN specification is later compiled into a binary file, when you want to run or test the CPN. The same CPN can be compiled into different binary files, depending on the compilation strategy chosen (see section 19.2). Since the CPN specification is in a textual format, you can edit it with your favorite text editor.

The causal probabilistic networks that form the knowledge bases used by the Hugin system are described by a special purpose programming language, called the CPN Language. Using this language, complete descriptions of CPNs can be created.

The fact that the CPN Language is a programming language does not mean that one has to be a programmer to use it. The CPN Language is a very simple language with only a few constructions.

The appendix describes how to create knowledge bases using the CPN Language. We first describe the files and directories used to store the specification and compiled CPN. Next, we describe how to invoke a text editor on a CPN specification. Finally, we describe the CPN Language itself. The description is given in terms of a number of examples. The final section of this appendix gives the detailed syntax of the CPN Language.

B.1 The Directory Structure

First we describe the directory structure used by Hugin Runtime. Start by moving to the directory you have specified in the CPN Browser (see section 10.1), say '`/user/john/Hugin`'.

```
mimer% cd /user/john/Hugin
mimer% ls
Domains  Specs
mimer%
```

This directory has two subdirectories:

Domains The '`Domains`' directory contains compiled CPNs. It includes a directory for each compiled CPN you have. Unless you are using the Hugin API, you will find little of interest in the '`Domains`' directory.

Specs The '`Specs`' directory contains the descriptions of all your CPNs. Each CPN description is stored in a separate file having the same name as the CPN itself.

The list of CPNs in the CPN Browser (see section 10.2) contains all the CPNs found in either directory.

The CPN Specification files has exactly the same names as used in the CPN Browser. Since these names often include space, you will have to quote the file name. To edit the Chest Clinic CPN with the '`vi`' editor, enter the following commands.

```
mimer% cd Specs
mimer% ls
Aunt Emily   Chest Clinic  Reproduction  Stud Farm
mimer% vi "Chest Clinic"
mimer%
```

B.2 Specifying Nodes

The basic components of a CPN are the *nodes* in the network. This is reflected in the CPN Language: A CPN is described in the CPN Language basically by describing the individual nodes of the network. For each one, the name of the node and its coordinates (used when drawing the network) must be given. The names of the states of the nodes and its parents, if any, must be specified . Finally, the potential tables for the node must be given.

Example B.1 A simple example of a node specification is as follows:

```
S   "Smoker?"                300    400
( "yes" "no" )
( 0.4 0.6 )
```

This declares a node named `S` with the label `Smoker?`. The position of the node is (300,400). The node has two states, named `yes` and `no` with a priori probabilities of 40% and 60%. This node has no parents in the CPN. ■

The name is used to identify the node and must be unique for the CPN, e.g. no other node named `S` must exist in the CPN containing the node from the example. Also, the characters used to make up the node names must be letters, numbers, or one of the characters `_/.-`. The first character must not be a number. Case is significant, `smoker`, `Smoker`, and `SMOKER` are all different names.

The label can be any string of characters except quote '"', and does not have to be unique. The label is the string that is printed on the node in the Base Window. The label should be something meaningful that can identify the node in the network.

The position of the node is where the node will appear in the graphic display of the CPN. The first number is the horizontal- or x-coordinate, while the second is the vertical- or y-coordinate, i.e. the distance from the left edge of the CPN and from the bottom of the CPN.

The origo of the coordinate system is in the bottom left corner of the display. The x-coordinate therefor measures the distance from the left-hand side of the Base Window to the lefthand side of the node. Likewise, the y-coordinate measures the distance from the bottom of the Base Window to the bottom of the node.

The above is not exactly true. It is modified as follows: If a large area of the CPN drawing is empty, for example if the lowest positioned node appears at y-coordinate 1000, the CPN drawing will be moved down. In other words, all y-coordinate will be decreased by the smallest y-coordinate used. The same applies for x-coordinates where the CPN is moved to the left.

This means that if you are in doubt about how high your network is going to be, you can start with y-coordinate 5000, say. This will provide you with ample space to grow in both directions.The same goes for the width of the CPN drawing.

After the position follows the list of states of the node. This is given as a opening parenthesis, a sequence of state names, and a closing parenthesis. The number of states of the node is determined from the length of the list. Internally, states are identified from their position in the list, e.g. state number 1, state number 2, etc.

The names of the states are used to identify the states in the bar charts of the monitor windows and in the Set state menu of the monitor. The state names should therefore be meaningful and be able to identify the states for display and input.

In the example, the node has no parents. We therefore simply specify the a priori probabilities for the states. These are given as a vector of values, with the first value corresponding to the first state, etc. In the example, The a priori probability for `yes` is 0.4 while the a priori probability for `no` is 0.6, e.g. 40% of the patients are smokers.

In the example, the probabilities in the vector add up to one. If this is not the case, the probabilities will be automatically normalized during compilation. When this happens, a warning is issued. It is good practice to enter the probabilities as normalized values. When this is done, warnings about automatically normalization can be used to catch typing errors in the tables.

B.3 Specifying Nodes with Parents

When the nodes have parents, the specification becomes a bit more complicated. The potentials must be given as a conditional probability table.

Example B.2 A node with a single parent is specified as follows:

```
L   "Has lung cancer"                200    300
( "yes" "no" )  S
(( 0.10 0.90 )         % S = yes
 ( 0.01 0.99 ))        % S = no
```

This declares a node named `L` with the label `Has Lung Cancer`. The position of the node is (200,300). The node has two states, named `yes` and `no`, and a single parent named `S`. ■

In example B.2, the parent S is specified following the list of states. The parent is specified by stating the name of the parent node.

The conditional probability table following the parent name in example B.2 is given as follows: For each state of the parent node, a vector of probabilities for the current node is given, e.g., if the node S is in its first state, the probability for L being in state yes is 0.10 and the probability for L being in state no is 0.90. If, on the other hand, the node S is in its second state, the probability for L being in state yes is 0.01 and the probability for L being in state no is 0.90.

If we take a look at the definition of node S, we see that this means the *if* a patient is a smoker, the risk of having cancer is 10%, whereas *if* a patient is *not* a smoker, the risk is only 1%.

In the example, note that the individual lines of the table are given a comment, starting with a % character. This is simply intended to help the human reader and is not used by the Hugin compiler. Also, the layout of the table is used only as an aid for the human reader. The compiler uses the parentheses to group the probability values. The parentheses must therefore be correctly nested.

If the parent of node L (node S) had more than two states, the potential table for L would have more than two probability vectors. For a node with one parent, the number of probability vectors will be equal to the number of states of the parent.

As can be seen from example B.2, comments are introduced by a % character and extend to the end of the line. Comments can be used at all places in the specification where whitespace is allowed. Beware that if you edit the CPN with Hugin Runtime, all your comments will be removed and replaced with comments generated by Hugin Runtime.

B.4 Nodes with several Parents

Nodes with more than one parent are specified in the same way as nodes with a single parent. The parents are stated in any order, and the potentials are given as a table ordered in the same way as the parents.

Example B.3 A node with two parents is specified as follows:

```
D   "Dyspnoea?"                  250    100
( "yes" "no" )  B E
((( 0.9 0.1 )           % B = yes, E = yes
  ( 0.7 0.3 ))          % B = yes, E = no
 (( 0.8 0.2 )           % B = no,  E = yes
  ( 0.1 0.9 )))         % B = no,  E = no
```

Again, the specification starts with the node name and label, the position, and the state name vector. This is followed by a list of names of the parents and finally the conditional probability table. ■

In example B.3 the node D has two parents B and E. The conditional probability table then gives the probabilities of the states of D for all combinations of the states of B and E. These are given as nested tables, in such a way that the states for the "last" parent varies "faster" than the preceding parents.

In the example, the conditional probability table for D consists of two parts: one for the case where the state of B is yes and one for the case where the state of B is no. Each part of the tables is again composed of two parts, governed by the states of E.

Note that the *two* above stems from the fact that each of the nodes B and E has two states. There should in general be a part for each states of the parent in question. This can be seen from the following example:

Example B.4 The node WA has two parents, A and B. Each parent has three states, 0, 1, and 2.

```
WA   "Mrs. Jones' Testimony"                250 700
("open" "nothing")       A   B
((( 10  90 )           % 0,  0
  ( 90  10 )           % 0,  1
  ( 90  10 ))          % 0,  2
 (( 90  10 )           % 1,  0
  ( 95   5 )           % 1,  1
  ( 95   5 ))          % 1,  2
 (( 90  10 )           % 2,  0
  ( 95   5 )           % 2,  1
  ( 95   5 )))         % 2,  2
```

We specify the node name, label, position, state names, and parents as usual. Then follows the potential tables, giving probability vectors for WA for each state combination of A and B. ■

Again, the potential table has a part for each state of the first parent (three parts in this case), and each part is split into probability vectors for each state of the second parent. Note that for each probability vector, the parent state combinations are given as comment under the specification of the parent names.

If a node has more than two parents, there will be additional levels of tables. On each level, the tables will have a number of parts, or *subtables*, corresponding to the number of states of the parent. Each subtable will then enumerate the combinations of the states of the remaining parents.

The following example, taken from the "Aunt Emily" example distributed with the Hugin system, demonstrates this:

Example B.5 The node D/A3 has three parents, D/A2, PA2, and PB2, and has two states Yes and No.

```
D/A3     "Alive Wednesday"                    250 420
("Yes" "No")              D/A2  PA2   PB2
((((  0.05  0.95 )       % yes,  yes,  yes
   (  0.10  0.90 ))      % yes,  yes,  no
  ((  0.10  0.90 )       % yes,  no,   yes
   (  0.99  0.01 )))     % yes,  no,   no
 (((  0.00  1.00 )       % no,   yes,  yes
   (  0.00  1.00 ))      % no,   yes,  no
  ((  0.00  1.00 )       % no,   no,   yes
   (  0.00  1.00 ))))    % no,   no,   no
```

Each parent has two states yes and no. For each combination of parent states, the probability of each state of D/A3 is given. ■

The potential table in example B.5 has three levels. On the first level, the table has two parts, one for node D/A2 being in state yes and one for node D/A2 being in state no. Each part is then split into further subtables, according to the states on node PA2, and so on.

B.5 The Complete Specification

A complete specification of a CPN in the CPN Language consists of the specifications of the nodes, stated in any order, plus a small amount of additional information. We illustrate this by giving the complete specification of a very simple CPN.

Example B.6 The complete Specification of the Burglary network. This network has only two nodes, the Burglary node (named B) and the Alarm node (named A), with B being the parent of A.

```
Burglary "Burglary"     % Domain name and label
 100 40                 % node dimensions (pt)
   1                    % scale factor

% Node no. 1

B "Burglary"        100 200
( "yes" "no" )
( 0.0001 0.9999 )


% Node no. 2

A "Alarm"                  100 100
( "yes" "no" )   B
(( 0.95 0.05 )       % B = "yes"
 ( 0.01 0.99 ))      % B = "no"
```

The network specification starts with the name and label of the CPN. Then follows the size of the nodes and the scale factor for the network. Finally, the individual node specifications are given. ■

In example B.6, we start the specification by stating the *CPN name* and the *CPN label*. The name is used to identify the CPN to the Hugin Inference Engine. Only one CPN with a given name can be loaded at a time. The characters used in the name follow the same rules as the characters used in node names. The CPN name and label are not currently used by Hugin Runtime, but they are available through the Hugin API.

Then follows the dimensions of the nodes in the network. The nodes of the CPN are drawn as boxes with the height and width given. These dimensions are stated in printer's points[1]. Using this facility, you can adapt the size of the nodes to the number of nodes and the layout of the CPN.

The final number in the CPN header is a *scale factor* for the network graphics. This number is used to scale the complete graphical representation of the network (both node sizes and positions). This is useful for resizing the networks without having to change all the individual coordinates.

The scale factor can also be used to change the unit of measure. If the scale factor is set to 28.34, the unit is effectively changed to centimeters, i.e., all node positions will be in centimeters.

Only the node sizes and node positions are affected by the scale factors. The line thickness and the font used are constant. See also section 19.4.

Often, small networks are drawn with large nodes and good distance between nodes. As more nodes are added to the network (i.e., more knowledge is added to the knowledge base), it will be necessary to scroll to see the complete network. When this is the case, it is often an advantage to scale down the graphical representation. This can be achieved by simply changing the scale factor.

The CPN specification header is followed by the specifications for individual nodes. The nodes can be given in any order, that is, nodes can be specified either before or after their parents. In many applications there will exist a logical grouping of the nodes. It is a good idea to use this grouping in the specification to make editing simpler and the specification simpler to grasp. The Hugin Inference Engine does not care about the order and will order the nodes internally as needed.

[1] 1 pt = 1/72 inch

B.6 Syntax of the CPN Language

This section specifies the syntax of the CPN Language. This section does not add anything new, it merely states precisely what has been said in the preceding sections.

In the following we will use the conventions that <... *name*> denotes an identifier, a nonempty sequence of letters, with the first element being a letter ('_', '-', '/', and '.' will be regarded as letters). We will also use <... *label*> to denote a string, a sequence of characters not including '"', surrounded by '"', e.g., `"A text"`.

The syntax for a CPN description is

<*CPN description*> → <*CPN declaration*>
<*node declaration*>*

The purpose of the <*CPN declaration*> is to specify general information for the complete CPN, such as the name of the CPN and the size of nodes. The syntax is

<*CPN declaration*> → <*CPN name*> <*CPN label*>
<*dimension information*>
<*dimension information*> → <*node width*> <*node height*>
<*scale factor*>

The <*CPN name*> is the name of the CPN, that is, the name of the knowledge base being defined. The <*CPN label*> is a text label associated with the knowledge base. The width of a node when displayed will be

<*node width*> * <*scale factor*> pt

where 1 pt is 1/72 inch. The height of the node will be found in the same way. The <*scale factor*> is also used to scale the coordinates used for positioning of nodes.

A node is described in the following way

<*node declaration*> → <*node name*> <*node label*>
<*node position*> (<*state label*>+)
<*parent node names*> <*potential table*>

The <*node name*> is the name used to reference a node when it is to be used as a parent of some other node. The node name must be unique. The <*node label*> will be used to identify the node in the CPN drawing. The <*node position*> specifies where the node will appear in a graphical representation of the CPN. The position is given as

<*node position*> → <*x-coordinate*> <*y-coordinate*>

The sequence of <*state label*> specifies the name of each state of the node and the sequence <*parent node*> specifies the names of the parents of the node, e.g.

<*parent node names*> → <*node name*>*

The potential table is given as a list, whose members must be either numbers or further (sub)tables, e.g.

<*potential table*> → ((<*number*>* || <*potential table*> *))

The length of the list and the level of nesting should match the number of states of the node and the number of parents and their states. For example, a node with two states and no parents can be given as

```
S "Smoking?"
150 180
("Yes" "No")
(0.50 0.50)
```

and a node with two states and one parent can be given as

```
L "Lung Cancer?"
100 120
("Yes" "No") S      % S is a parent
((0.10 0.90)
 (0.01 0.99))
```

In this example, 0.10 is the probability of state “Yes” of node L, given that node S is in state “Yes”. Likewise, 0.90 is the probability of state “No”of node L, given that node S is in state “Yes”, and so forth.

C

Other Fileformats

C.1 The Numbering File

The numbering file is simply a list of node names with one node per line. Whitespace and everything following a per cent sign is ignored. The numbering for a CPN named `foobar` can be found as:

```
$HUGINHOME/Domains/foobar/foobar.num
```

An example numbering file for the Aunt Emily CPN (see section A.3) generated by Hugin Runtime is listed below.

```
%
% Node numbering for: /usr/hugin/H/Specs/AuntEmily
%
% Thu Apr  9 13:23:05 1992
%

M               % Murderer
O               % Autopsy
WB              % Dora's Testimony
WA              % Mrs. Jones' Testimony
LB2             % S Left Tuesday
LA2             % H Left Tuesday
LB1             % S Left Monday
CD              % Cause of Death
D/A3            % Alive Wednesday
PB2             % S Taken Tuesday
PB1             % S Taken Monday
PA2             %  H Taken Tuesday
D/A2            % Alive Tuesday
B               % Sedative (5)
LA1             % H Left Monday
PA1             % H Taken Monday
A               % Heart Drug
```

C.2 The Cases Files

A case file is also a list of node names with one node per line. Each node name is followed by a list of state names. The list is surrounded by parentheses, and the state names are surrounded by double quotes. Whitespace and everything following a per cent sign is ignored. The case files can be found in the following directory:

```
$HUGINHOME/Cases
```

A small example generated by Hugin Runtime containing two ordinary findings and one multistate finding is listed below.

```
%
% Findings for: AuntEmily
%
% Mon Apr 13 16:51:27 1992
%

A       ( "2" )               % Heart Drug
B       ( "1" "2" )           % Sedative (5)
D/A2    ( "No" )              % Alive Tuesday
```

C.3 The Adaptation File

The adaptation file format is intended to replace most other existing and future file formats used by Hugin Runtime, and is therefore a bit more complicated. The format basically consist of a list of objects to be modified, and for each object, a list of attributes to be set. Currently the only objects which can be modified are the nodes in the CPN, and the only setable attributes are the ess and fading tables.

The tables has one entry for each parent table combination. The content of the ess table is exactly the same as the sample sizes entered in the Node Properties Window. The fading table has multiple uses:

- A 0.0 indicates that adaptation is not enabled for the parent state combination.
- A 1.0 indicates that adaptation is enabled, but fading is not.
- A number *f* between 0.0 and 1.0 indicates that both adaptation and fading is enabled, with the maximum sample equivalent to `1.0 / (1.0 - f)`.
- A negative number indicates that this parent state combination had adaptation enabled, but it was later discovered that there was only one positive state, thus making adaptation impossible.
- There should never appear numbers larger than 1.0.

A small example generated by Hugin Runtime i listed below.

```
%$ HUGIN 1.2 Adaptation information for "AuntEmily".
% Generated by Hugin Regular 3.0.

node $WB =
{};

node $WA =
{
```

```
    ess = (( 10 10 10 )
           ( 10 10 10 )
           ( 10 10 10 ));

    fading = (( 1 1 1 )
              ( 1 1 1 )
              ( 1 1 1 ));
};
```

Each node to be modified is introduced by the keyword `node`. The name of the node is prepended with a dollar sign. For the node `WB` no attributes is changed. The declaration

```
node $WB =
{};
```

has no effect.

For the node `WA`, both the `ess` and the `fading` tables are set. The format of the tables themselves are the same as used by the net files, described in appendix B.

D

OPEN LOOK

The OPEN LOOK GUI is a graphical user interface standard developed by AT&T and Sun (see [Sun89]), based on technology licensed from Xerox. The idea behind OPEN LOOK is that you, as a user, should be able to easily use any OPEN LOOK application once you have learned one OPEN LOOK application.

OPEN LOOK is a trademark of AT&T, and although we have tried to obey the OPEN LOOK specification when possible, we have not yet tried to get Hugin Runtime certified as OPEN LOOK compliant.

This appendix serves two purposes. First, we introduce the OPEN LOOK terminology used throughout the manual. Second, we have omitted a number of features in the description of Hugin Runtime, because they really are generic OPEN LOOK features, and not specific to Hugin Runtime. We have placed these descriptions in an appendix, because users who already know OPEN LOOK will not need to learn it again.

This appendix will *not* introduce window based system in general. You are assumed to know the basic concepts behind graphical user interfaces, such as windows, mice, and icons.

D.1 The Concepts behind OPEN LOOK

OPEN LOOK builds on a very limited set of concepts, which it can be useful to remember when using an OPEN LOOK application. First, an OPEN LOOK application manipulates a number of objects. In Hugin Runtime, these objects are:

- Nodes and links between nodes.
- CPNs, containing nodes and links.
- The application (tool) itself.

OPEN LOOK provides two different ways to manipulate these objects. The first is *direct manipulation*, where you directly manipulate the objects with mouse. Direct manipulation is used for operations that are both common and simple. Hugin Runtime uses direct manipulation to move nodes in the Base Window, and to create nodes and links.

The second way to manipulate objects is more indirect. First you *select* the object using the mouse, and then you *operate* on the objects by activating a control. An example from Hugin Runtime is deletion of nodes and links, where you first select the nodes and links you want to get rid of, and then activate the Delete Item from the Edit Menu.

A special case of select-then-operate is setting the properties of a window. Most objects have some properties which the user can alter. This is normally done by selecting the object, and then activating the appropriate item from the Properties Menu. Hugin Runtime allows you to alter the properties of an individual node, a complete CPN, and the tool itself. As you can see from chapter 13, chapter 12, and chapter 19 property windows has a common layout with the settings in the top of the window, and Apply and Reset Buttons (and optionally a Set Defaults Button) in the bottom of the window. See the three chapters for details.

D.2 The Mouse

OPEN LOOK relies heavily on the use of a mouse, preferable one with three buttons[1]. The three mouse buttons each has a specific purpose and name.

SELECT Usually the left mouse button. Used for selecting objects and activating controls. For example, you select a node by clicking (press and release) SELECT while the mouse pointer is above the node. Similarly, you activate a button by clicking SELECT on it.

ADJUST Usually the middle mouse button. Used for adjusting the selection. If you click ADJUST on an unselected node, the node is added to the selection. Similarly, if you click ADJUST on a selected node, the node is removed from the selection.

MENU Usually the right mouse button. Used for displaying a menu. If you press MENU on a menu button, a new menu is displayed. You can now activate an item on the menu by dragging (i.e. moving the mouse without releasing the mouse button) the mouse pointer to an item, and then releasing MENU.

If the mouse connected to your workstation only has two or less buttons, the ADJUST (and MENU) buttons will be bound to keys on the keyboard. Refer to local documentation for details.

When you move the mouse, the *mouse pointer* will follow it on the screen. We say that you *move* the mouse pointer *on* or *above* items on the screen by moving the mouse on your table. We use the following terminology for using the mouse. We say that you *press* and *release* mouse buttons. If you press and release a mouse button without moving the mouse, we say that you *clicked* the mouse button. If you press a mouse button and then move the mouse while holding the button down, we say that you *drag* the mouse pointer.

D.3 The OPEN LOOK Controls

The OPEN LOOK GUI describes a number of standard controls for activating operations and modifying objects. In this section we will introduce the different types of OPEN LOOK controls used by Hugin Runtime.

[1] If you wonder why OPEN LOOK has to do with small fury animals with multiple buttons, you should probably read the beginners guide to you workstation again.

D.3.1 Buttons

The simplest control is the *button*. Not mouse buttons, but buttons drawn on the screen. The Insert Button on the CPN Browser is an example of such a control, as illustrated by figure D.1.

Figure D.1. *The Insert Button*

You *activate* a button by moving the mouse pointer above the button, and then clicking SELECT. Activating the button will cause the operation associated with the button to be performed, in this case the current directory will be inserted to the directory menu (see section 10.1).

Some buttons have an menu associated with them. They are distinguished from ordinary buttons by a *menu mark*, the small triangle illustrated by figure D.2.

Figure D.2. *The File Menu Button*

The figure illustrates the File Menu Button on the Base Window. You can display the associated File Menu by pressing or clicking MENU on the File Menu Button, as illustrated by figure D.3.

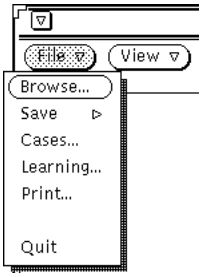


Figure D.3. *The File Menu*

How to activate the different items on the menu will be described in the next section.

As can be seen, the Browse Item is surrounded by a thin oval. This is because the Browse Item is the default item in the File Menu. If you activate a menu button by clicking SELECT, the operation associated with the default item will be performed. It is possible to change the default item by using the CONTROL modifier key on the keyboard. If you hold down the CONTROL key while trying to activate an item on a menu, you will set the default item. For example, to make the Quit Item default for the File Menu (not very useful!), click MENU on the File Menu Button, move the mouse pointer to the Quit Item, press the CONTROL key, click SELECT, and release the CONTROL key. You can now leave Hugin simply by clicking SELECT on the File Menu Button.

Since changes to the default item will not be remembered next time you start Hugin, this is a rather useless example. However, if you are performing the same operation many times, the ability to change the default item can be very convenient.

Hugin Runtime also use a special variant of menu buttons, called *abbreviated menu buttons*. These are exactly equivalent to ordinary menu buttons, except that the button itself is abbreviated into a square box with a menu mark. These are often used in Hugin Runtime together with a text field, as illustrated by figure D.4.

The CPN Menu from the CPN Browser (see section 10.2) gives access to a number of operations which can be performed on the selected CPN.

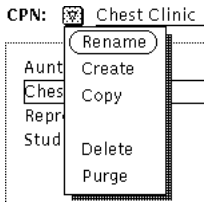


Figure D.4. *The CPN Menu*

D.3.2 Menus

You can display a menu with the MENU mouse button in many different areas of the screen, not just on the menu buttons. As an example, pressing MENU on the Base Window Pane will display a pop-up menu[2] with the most common operations, as illustrated on figure D.5.

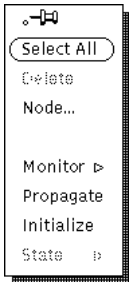


Figure D.5. *The Pane Menu*

As you can see, the four last menu items are *dimmed*. Generally, if you for some reason can not activate a control right now, it will be dimmed. If you press HELP on a dimmed control, a help window (see 20.1) explaining why the control is dimmed will appear.

There exist two different methods for activating menu items. One way is first to click MENU to make the menu appear, then move the mouse pointer to the item you want activated, and finally click MENU to activate the item. The other way is to press MENU to make the menu appear, then drag the mouse pointer to the item you want activated, and finally release the MENU mouse button to activate the item. Use the method which suits you best.

Just like there exist different sorts of buttons, there also exist different sorts of menu items. Ordinary menu items are just like plain buttons, activate them in order to perform some action.

Some menu items has a submenu associated with them. These menu items are marked with a small triangle. You can display the submenu by clicking MENU on the item, or by dragging the mouse pointer to the right. If you click SELECT on a menu item, you activate the default item in the submenu. The default item can be changed in the same way as described in section D.3.1 under menu buttons.

Finally, some menu items has an window associated with them. They are marked with three dots. When you activate such an item, the window will be displayed. If the window is already displayed, it will be updated and put to the front of the screen. The File Menu in the Base Window contains all three kinds of items, as illustrated by figure D.3. Figure 9.6 illustrates the Save Menu as well.

[2]It is called a pop-up menu because it does not have an associated menu button.

D.3.3 Exclusive Settings

In some situations you are allowed to choose between a fixed number of possibilities. For such situations, OPEN LOOK offer an exclusive setting control. The most prominent exclusive setting in Hugin Runtime is the Mode Setting in the base window, illustrated on figure D.6.

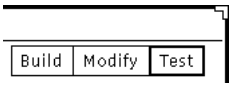


Figure D.6. *An Exclusive Setting*

You choose the setting by clicking SELECT above it. To switch to Edit mode, click SELECT on Edit. The chosen setting is emphasized with a thick border.

In some places the exclusive setting is hidden in an abbreviated menu button. An example of this is the Parent State item in the Node Properties window. If there are too many states to fit an exclusive setting, an abbreviated menu button will be used instead. You choose a state in the same way as you would activate an item in an ordinary menu.

D.3.4 Scrollbars

When it is necessary to display an unknown amount of information, OPEN LOOK provides a scrollbar to browse through it. Scrollbars are used by the scrolling lists (see section D.3.5) and the Base Window Pane (see section 9.1). Scrollbars are used for moving a small fixed sized window to a larger, potentially variable sized, object.

A scrollbar consists of of several elements (a horizontal scrollbar is shown in figure D.7). First of all we have the *cable* (the long "string"). At each end of the cable we have *cable anchors* (the small boxes), and on the cable the *elevator* is located. The elevator consists of two *arrows* and a *drag area*.

Figure D.7. *An Horizontal Scrollbar*

The location of the elevator reflects the selected view of the display area. If the elevator is positioned at the top (or left), the top (left most) part of the object is displayed. If the elevator is placed at the bottom (or right), the bottom (right most) part of the object is displayed.

The color of the cable also gives some information. The dark area of the cable corresponds to the portion of the object that is currently visible.

The view can be changed in several ways.

- *Go to the beginning of the display area*. Click SELECT on the top/left cable anchor.
- *Go to the end of the display area*. Click SELECT on the bottom/right cable anchor
- *Move the view one unit up/left.* Click SELECT on the up/left arrow on the elevator.
- *Move the view one unit down/right*. Click SELECT on the down/right arrow on the elevator.
- *Move the view one page up/left.* Click SELECT on the cable above/at the left of the elevator.
- *Move the view one page down/right.* Click SELECT on the cable below/at the right of the elevator.

- *Move to an arbitrary location*. Press SELECT on the drag area, and keep the button pressed. Then drag the elevator to the desired location and release the button.

The scrollbar also has a pop-up menu. The *scrollbar menu* can be accessed by pressing MENU on the scrollbar area. (see figure D.8). The menu is used to position the view of the display area or, for the Base Window Pane only, splitting the view.

Figure D.8. *The Pane Scrollbar Menu*

On the menu of the vertical scrollbar you find ‘Here to top’, ‘Top to here’, ‘Previous’, and ‘Split View’. On the horizontal scrollbar the first two items are labeled ‘Here to left’ and ‘Left to here’.

Here to top (Here to left) Used for locating the view so that the selected item becomes the top (left) in the view of the display area.

Top to here (Left to here) The opposite of the previous. The top (left) of the display area is positioned at the selected position.

Previous The previous view of the display area is always remembered. The Previous entry in the scrollbar menu is used to replace the current view with the previously selected view.

Split View The Split View Item does not appear in scrolling lists, and will be discussed in section D.4

D.3.5 Scrolling Lists

A scrolling list is a window to a potentially large list of text items. The window is controlled by a scrollbar, as described in the previous section. The CPN Browser window (see figure 10.1) is dominated by a single scrolling list, where you can select the CPN you want to work with. You select an item in a scrolling list by clicking SELECT on it. You deselect an item by selecting any other item, or by clicking SELECT on the already selected item. There can be at most one selected item per scrolling list in Hugin Runtime[3].

There is a menu attached to the scrolling lists. You can display the menu by clicking the MENU mouse button. As illustrated by figure D.9, the menu has a single item, namely Locate Choice. Activating Locate Choice will scroll the list in order to ensure that the selected item is displayed.

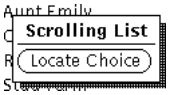


Figure D.9. *The Scrolling List Menu*

[3]OPEN LOOK allows multiple selected items, so scrolling lists in other OPEN LOOK applications may behave differently.

D.3.6 Check Boxes

Check boxes (see figure D.10) are used for enabling or disabling a specific feature. Click SELECT on the check box to toggle between enabled and disabled. When a check box is enabled, a check mark is displayed. On figure D.10 the Compress Item is enabled, but the Approximate Item is disabled.

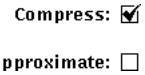


Figure D.10. *Two Check Boxes*

D.3.7 Sliders

The Node Properties Window allows you to adjust the probability of a state by using a *slider*, or by typing into a text field. The slider is illustrated by figure D.11.

Figure D.11. *A Slider*

The slider consist of a bar with a drag box. The position of the drag box illustrates the relative probability of the state. If the drag box is located to the left of the bar, the probability is small. Similarly, if the drag box is located to the right, the state has a high probability. In order to adjust the state, press SELECT on the drag box, and drag it left or right in order to decrease or increase the probability, respectively. You can also click SELECT on the slider bar at either side of the drag box for fine grained adjustment of the state.

D.3.8 Text Fields

A text field is simply a control which allows you to type in textual information from the keyboard. In order to activate a text field, simply click SELECT in it. Input from the keyboard will now be directed to the text field. You can enter text simply by typing letters on the keyboard. The text will be inserted at the location of the small triangular cursor. You can move cursor either by clicking SELECT at the new position, or by using the cursor keys located at the right side of the keyboard. Typing DELETE will remove the character to the left of the cursor.

The three special keys COPY, PASTE, and CUT[4] can be used for manipulating larger pieces of text.

To cut a piece of text, you must first select it. You can select the text by pressing SELECT at the beginning of the text, dragging the pointer to the end of the text, and releasing SELECT. The selected text will be highlighted, as illustrated on figure D.12, where the word ‘john’ is selected. You can also select a single word by double clicking SELECT on it. When you have selected the text, remove it by pressing the CUT key.

Figure D.12. *A Text Selection*

In order to move the text, cut it as described above, click SELECT at the destination, and press the PASTE key. The destination may be another place in the same text field, another text field in Hugin Runtime,

[4]The three keys can be found on L6, L8, and L10, respectively, on Sun keyboards. Read the documentation for other systems.

or even another OPEN LOOK application. This technique can also be used for moving text from other OPEN LOOK applications to Hugin Runtime.

If you want to copy the text instead of moving it, do as above, but press COPY instead of CUT.

A special sort of text fields are the *numeric text fields*. The numeric text field consist of a normal text field followed by an increment button and a decrement button, as illustrated by figure D.13. You can increment and decrement the numeric value by activating the appropriate buttons.

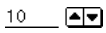


Figure D.13. *A Numeric Text Field*

D.4 Splitting a Pane

Using the Base Window Pane has been discussed in chapter 9. In this section we will discuss a nice feature of OPEN LOOK, namely the ability to split a view.

It is possible to split the Base Window Pane into several views of the CPN, as illustrated by figure D.14.

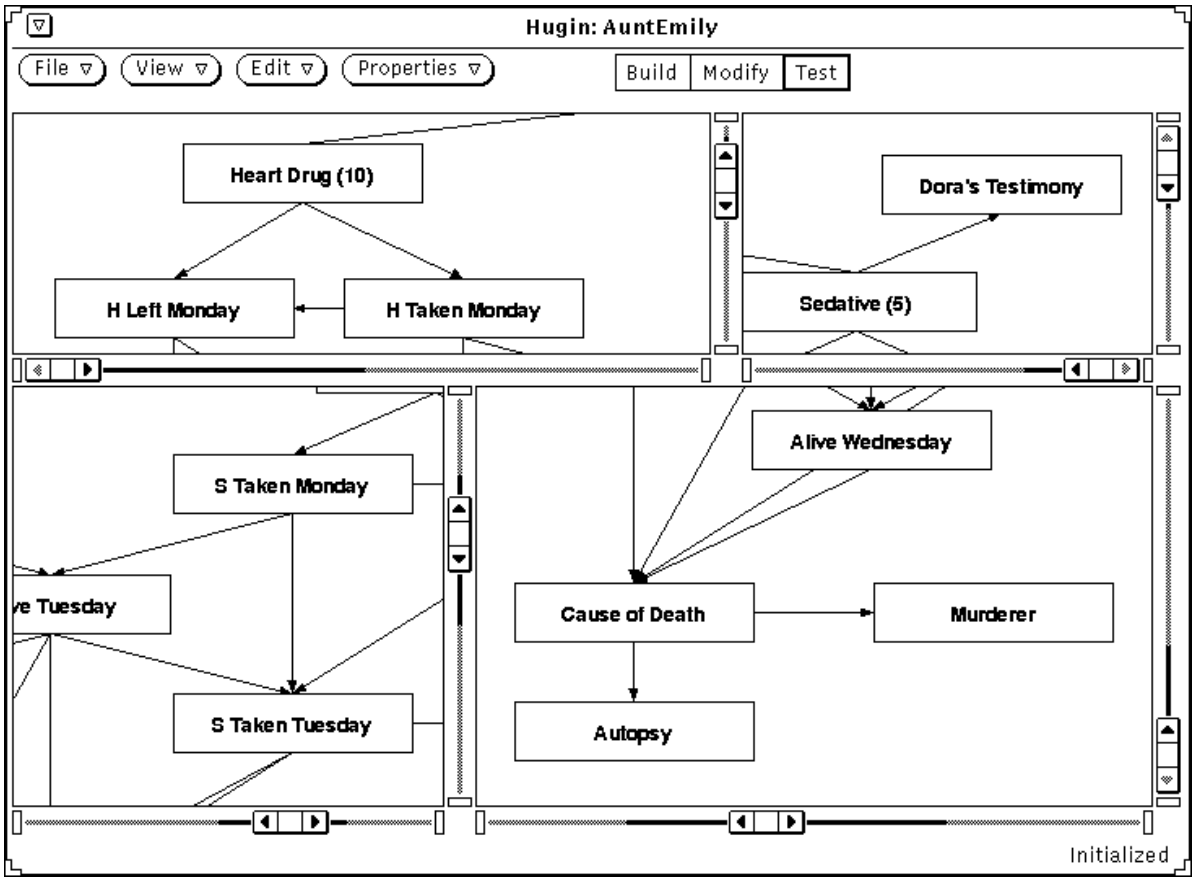


Figure D.14. *Multiple Views of the Aunt Emily CPN*

To split the view, activate the Split View item from the scrollbar menu at the place where the view should be split. Each of these views will act the same as just one single view; each view having their own scrollbars.

Another way of splitting the view is using the cable anchors. Press the SELECT in the one of the cable anchors, then drag to the position where the split should be and release the SELECT button. A line will follow the mouse as you perform this operation.

Splitting the domain window enables you to have several views of the CPN. This is useful in situations where the CPN is large and even a resizing of the window to fit the whole screen will not show the whole CPN. In such a situation splitting the view will enable you to see “both ends” of the CPN simultaneously.

When the view has been split an additional entry has been attached to the scrollbar menu, namely ‘Join Views’ (see figure D.15). Activating this menu item will join the view again.

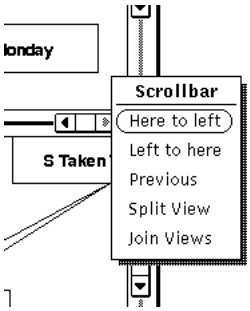


Figure D.15. *The Join Views Menu Item*

D.5 Window Scaling

While we are discussing the nice features of OPEN LOOK, window scaling is among the better ones. If you find the windows too small, you can start Hugin Runtime with the “`-scale large`” or “`-scale extra_large`” option. Figure D.16 illustrates an extra large CPN Properties Window. Compare the text in the window with the window header.

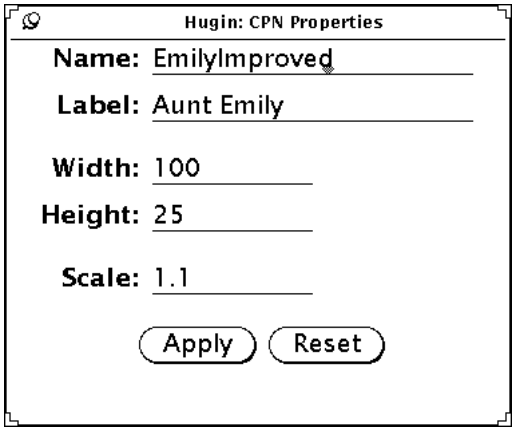


Figure D.16. *An Extra Large CPN Properties Window*

As can be seen from the figure, the “`-scale`” option affects the size of all the OPEN LOOK controls, but not the windows decoration. The contents of the Base Window Pane and the monitor windows are not affected, but can be specified from the Tool Properties Window (see section 19.4 and section 19.3).

You can also start Hugin Runtime with smaller windows, as illustrated by figure D.17, by using the “`-scale small`” option.

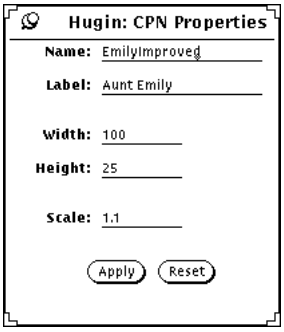


Figure D.17. *A Small CPN Properties Window*

If you compare figure D.17 with figure D.16, you will notice that the size of the window decoration (header, resize corners, and push-pin) stays the same, while all the controls are half the size. You can try for yourself, by typing the following commands:

```
mimer% xhugin -scale small &
mimer% xhugin -scale extra_large &
```

D.6 The OPEN LOOK Window Manager

This section describes how the most common window operations is performed with OPEN LOOK Window Manager. These operations depends on what window manager you are running. To see a discussion of the different window managers, read appendix E.

D.6.1 Moving a Window or Icon

Press SELECT in the window header, and drag the window to a new position. Hold down CONTROL to restrict the movement to a single direction. Hold down SHIFT to see the window move while dragging it. Hold down META to make minor corrections to the position.

You can move multiple windows simultaneously by selecting them first. You select the windows by clicking SELECT on the first window, and ADJUST on the other windows[5]. Selected windows are emphasized with an extra thick border, as illustrated by figure D.18. If you now try to move one of the selected windows (as described above), all the selected windows will move. This is particularly useful for moving the monitor windows together with the Base Window.

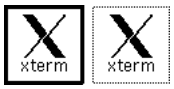


Figure D.18. *Two icons, where the one to the left is selected.*

D.6.2 Resizing a Window

Press SELECT in one of the resize corners (see figure D.19), and drag the corner. Release SELECT when you are satisfied with the window size. Hold down CONTROL to restrict the residing to a single direction.

[5]Yes, you select windows using the mouse as described in section D.2.

Figure D.19. *A Resize Corner*

D.6.3 Closing a Window and Opening an Icon

Click SELECT on the small triangle in the upper left corner of the base window. The base window and all pop-up windows will be collapsed into a single icon, as illustrated by figure D.20. The name of the current CPN is shown in the bottom of the icon.

Figure D.20. *The Hugin Regular Icon*

Double click SELECT on the icon. The base window and all pop-up windows will be restored to their original size.

D.6.4 Dismissing a Pop-up Window

Click SELECT on the push-pin in the upper left corner of the window. The push-pin is illustrated by figure D.21.

Figure D.21. *A Push-pin*

To dismiss all pop-up window, first click MENU on the header of a pop-up window to display the pop-up window menu. Then click MENU on the Dismiss Item to display the Dismiss Menu. Finally, click SELECT on the All Pop-ups Item. The menus are illustrated on figure D.22.

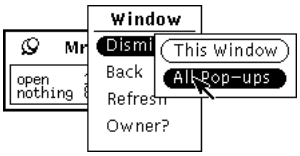


Figure D.22. *The Dismiss All Pop-ups Menu Item*

E

The X Window System and Window Managers

The X Window System does not include a policy for handling such matters as window placement and sizing, and does not include a user interface for moving and resizing window. Instead, this type of *window management* operations are delegated to a separate application, called the window manager.

The window manager is an X application like any other application. In general it includes a user interface that lets the user move windows and icons about on the screen, resize windows and so on. It typically also includes facilities for starting applications from a menu, etc. A window manager will also include a mechanism for negotiating window locations and window sizes with other applications.

It is important to realize that the window manager assumes the ultimate responsibility for placing windows on the screen. This means that it is the window manager that generates the *borders* around the windows and it is also the window manager that decides where a window will be located.

Throughout this manual we have been using the OPEN LOOK Window Manager (`olwm`), the default window manager from Sun Microsystems. As window borders, window close buttons, border menus, window border pushpins, etc., are all properties of the Open Look Window Manager, these items will be different if other window managers are used. Note that only the border of the window will change. The interior of a window should remain consistent across window managers.

A great number of window managers exist, often in many variants. This means that it is impossible for us to have tested Hugin Runtime with all possible window managers. We test Hugin Runtime with the major ones, including OPEN LOOK Window Manager (`olwm`), Motif Window Manager (`mwm`, see figure E.1), and Tab Window Manager (`twm`, see figure E.2). We recommend that you use `olwm`, as this will give you a consistent user interface, compliant with the OPEN LOOK user interface specification.

E.1 Window Placement

It is the job of the window manager to decide where a window will appear on the screen and what size the window will have. The application can supply *hints* to the window manager, but the window manager is

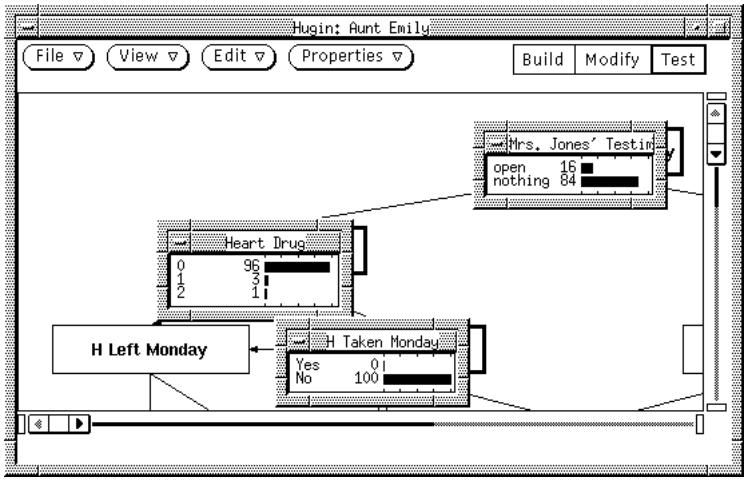


Figure E.1. *The Motif Window Manager*

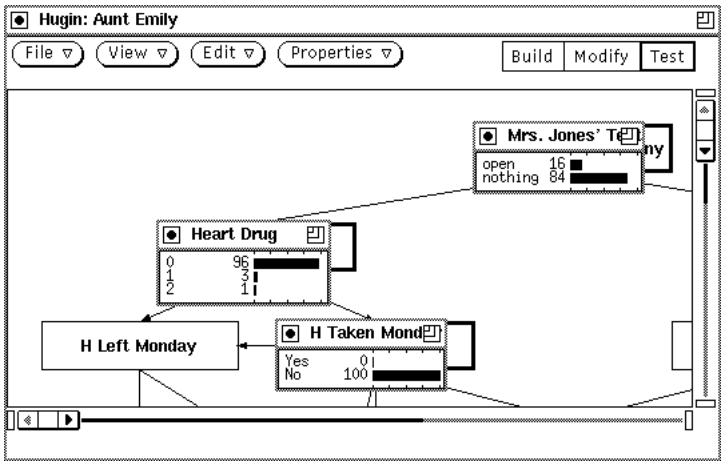


Figure E.2. *The Tab Window Manager*

free to override these hints, e.g. to enforce a specific policy on window placement and sizing.

This has some implications for the workings of Hugin Runtime, especially concerning monitor windows. Hugin Runtime tries to created the monitor window at the location of the node on the Base Window, but as it is the window manager which decides where the window appears, we cannot guarantee that this will work. When the OPEN LOOK Window Manager is used, the monitor window appears with the upper left corner at the location of the node in the . With other window managers, this might not work. The Motif Window Manager places monitor windows on a line from the top left to the bottom right corner of the domain window. The Tab Window Manager may allow you to place the monitor windows yourself, depending on the configuration.

E.2 Icons and Closed Windows

It is the job of the window manager to handle icons and closed windows. The OPEN LOOK Window Manager iconifies a window when closed, and closes all subwindows. The Tab Window Manager builds an *icon dock* that includes all windows, including the monitor windows. You can configure `twm` to use

ordinary icons instead of an icon dock. The Motif Window Manager does also create an (oridinary) icon for each window.

How to close a window varies with the window manager. All window managers have window borders with close boxes that you click to close the window. The close box is located at the left side if you are using the OPEN LOOK Window Manager or the Tab Window Manager, and at right side if you are using the Motif Window Manager.

E.3 Pinned Windows

Hugin Runtime uses the OPEN LOOK feature of *pinned windows* for command frames and property windows. A pinned window is an ordinary window with a *pushpin* instead of a close box in the window header.

For pushpins to work, the window manager must support them, as the pushpin is part of the window border, generated by the window manager. Currently, the only window manager supporting pinned windows is `olwm`. If you want to take advantage of the pinned windows you must therefore use the OPEN LOOK Window Manager.

BIBLIOGRAPHY

.....

Bibliography

[AOJJ89] Stig Kjær Andersen, Kristian G. Olesen, Finn V. Jensen, and Frank Jensen. Hugin — a shell for building bayesian belief universes for expert systems. In *Proceedings of the Eleventh International Congress on Artificial Intelligence*, pages 1080–1085, August 1989. Reprinted in Glann Shafer and Judea Pearl: Readings in Uncertainty, Morgan Kaufman, 1990.

[FA91] Lars Peter Fischer and Per Abrahamsen. *Hugin API Reference Manual*. Hugin Expert A/S, Niels Jernes 10, DK-9220 Aalborg, Denmark, revision 1.0 for hugin api 1.2 edition, July 1991.

[JA90] Frank Jensen and Stig Kjær Andersen. Approximations in bayesian belief universes for knowledge-based systems. In *Proceesings of the Sixth Conference on Uncertainty in AI*, July 1990.

[JCNJ91] Finn V. Jensen, Bo Chamberlain, Thorsten Nordahl, and Frank Jensen. Analysis in HUGIN of data conflict. *Uncertainty in Artificial Intelligence*, 6:519–528, 1991.

[JLO90] Finn Verner Jensen, Steffen L. Lauritzen, and Kristian G. Olesen. Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, pages 269–282, 1990.

[Kjæ90] Uffe Kjærulf. Triangulation of graphs — algorithms giving small total state space. Technical Report R 90-09, Institute of Electronic Systems, Department of Mathematics and Computer Science, University of Aalborg, Frederiks Bajersvej 7, DK-9220 Aalborg, Denmark, March 1990.

[LS88] Steffen Lilholt Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.

[OLJ92] Kristian G. Olesen, Steffen L. Lauritzen, and Finn V. Jensen. ahugin: A sytem creating adaptive causal probabilistic networks. draft for later article, February 1992.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems — Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, California, 1988.

[SL90] David J. Spiegelhalter and Steffen L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *NETWORKS*, 20:579–605, 1990.

[Sun89] Sun Microsystems, Inc., 2550 Garcia Avenue. *OPEN LOOK Graphical User Interface — Functional Specification*, 1989.

[Sun90] Sun Microsystems, Inc. *OpenWindows Version 2 User's Guide*, 1990.